

THE BIOMECHANICS
OF
BONE TURNOVER

A Dissertation

Presented to the Faculty of the Graduate School
of Cornell University

In Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy

by

Craig R Slyfield Jr.

January 2012

© 2012 Craig R Slyfield Jr.

All Rights Reserved

THE BIOMECHANICS OF BONE TURNOVER

Craig R. Slyfield Jr., Ph. D.

Cornell University 2012

Osteoporosis is a disease characterized by low bone mass and increased fracture risk. In adults, bone mass is primarily modified through bone remodeling. Bone remodeling is the coordinated activity of osteoclasts resorbing bone and osteoblasts forming new bone. Bone remodeling occurs at discrete locations on bone surfaces. The number and size of individual remodeling events influence the total amount of bone turnover in the body. Bone turnover is associated with increased fracture risk independent of bone mass. It is therefore believed that the number and size of individual remodeling events are important factors related to bone fragility and fracture risk. Precisely how metabolic bone disease alters individual remodeling events is not known. As a result, the mechanisms behind how metabolic bone disease alters bone biomechanics to increase bone fragility and fracture risk have not been determined. In the current dissertation, a sub-micron resolution three-dimensional imaging technique was developed to visualize individual remodeling events. The ability to measure individual remodeling events provides a means to understand how alterations in bone biology result in changes in bone turnover and fracture risk. This capability also allows for interpretation of individual remodeling events in terms of basic cell functions (proliferation, differentiation, motility, etc.). Using an animal

model of postmenopausal osteoporosis, the serial milling approach was then applied to study how estrogen depletion alters bone remodeling at the level of individual remodeling events. Finally, the serial milling approach was used to determine how bone remodeling is related to bone biomechanics. The techniques developed in the current dissertation provide a means of understanding how metabolic bone disease increases fracture risk by altering individual remodeling event number and size. The current dissertation suggests that estrogen depletion primarily influences osteoclast proliferation and differentiation. Therefore, the increase in bone turnover in postmenopausal osteoporosis is attributed to an increase the number of resorption cavities and, potentially, in the number of stress concentrations. Furthermore, the current dissertation also shows that resorption cavities are preferential sites of microscopic tissue damage formation in cancellous bone. Together, these results suggest a potential mechanism behind increased fracture risk associated with increased bone turnover.

BIOGRAPHICAL SKETCH

Craig (CJ) was born in San Jose, California. He received a Bachelors of Science in Mechanical Engineering from the University of California, Berkeley in 2006. From 2000 to 2006, he conducted research at the NASA Ames Research Center in Moffett Field, California. He focused on developing virtual reality training environments for rehearsing and developing medical procedures as well as biological experiments to be conducted on the international space station. He also developed a simulation platform for optimizing a biologically-inspired automated navigational control for an 8-legged Mars rover concept. During his final year at UC Berkeley, CJ conducted research to experimentally validate fracture risk predictions from quantitative computed tomography-based finite element models of human vertebral bodies. In August 2006, CJ began his graduate studies in mechanical engineering with an emphasis on biomechanics at Case Western Reserve University in Cleveland, Ohio. There, he advanced a serial milling approach to three-dimensional imaging of cancellous bone. The serial milling technique achieved sub-micron resolution with no limitation on physical specimen size. The technique was also capable of imaging fluorescent labels of bone formation and microscopic tissue damage. This work was ultimately published in the Journal of Microscopy and Bone. After conferral of his Masters of Science Degree, he began studies towards the degree of doctorate of philosophy in mechanical engineering with a specialty in biomedical and solid mechanics at Cornell University. There, he developed techniques to quantify the morphology of individual sites of bone resorption, formation, and microscopic tissue damage in cancellous bone. The techniques he and his team of fellow researchers developed can be applied to understand the mechanisms behind changes in clinical fracture risk due to metabolic bone diseases as well as drug therapies.

ACKNOWLEDGMENTS

I would first like to thank my parents. Without their encouragement and continued support, my academic achievements would not have been possible. I would also like to thank my team of research assistants. Their contributions were invaluable in developing the cutting edge techniques outlined in the current dissertation. I also thank the graduate advisory committee. The committee provided guidance that helped me to appreciate and articulate the broader impact of my contributions to the field of bone biomechanics. Finally, I would like to acknowledge the National Institutes of Health for providing funding for the research outlined in the current dissertation.

TABLE OF CONTENTS

Biographical Sketch	iii
Acknowledgements.....	iv
Table of Contents	v
List of Figures	viii
List of Tables.....	xii
1. Introduction.....	1
1.1. Motivation.....	1
1.2. Outline.....	4
2. Bone.....	6
2.1. Bone Function and Structure.....	6
2.2. Cancellous Bone Mechanics	8
2.2.1. Damage in Cancellous Bone	9
2.3. Osteoclasts and Bone Resorption.....	10
2.4. Osteoblasts and Matrix Production.....	12
2.5. Bone Remodeling	14
2.6. Regulatory Factors Involved in Bone Remodeling	16
2.7. Osteoporosis.....	18
3. Bone Histomorphometry	21
3.1. Static Histomorphometry	21
3.2. Dynamic Histomorphometry	26
3.3. Limitations of Traditional Two-Dimensional Histomorphometry	28
4. Serial Milling with Epifluorescence Imaging	31
4.1. Abstract.....	31

4.2. Introduction.....	32
4.3. Methods.....	36
4.3.1. Imaging System.....	36
4.3.2. Image Processing Overview.....	38
4.3.3. Non-uniform Illumination Correction.....	39
4.3.4. Mosaic Tiling.....	41
4.3.5. Sub-surface Signal Removal.....	42
4.3.6. Segmentation.....	44
4.4. Results.....	48
4.5. Discussion.....	51
4.6. Acknowledgements.....	55
5. Three-dimensional Dynamic Bone Histomorphometry.....	53
5.1. Abstract.....	55
5.2. Introduction.....	57
5.3. Methods.....	60
5.3.1. Image Acquisition.....	60
5.3.2. Image Processing and Validation.....	61
5.3.3. Three-dimensional Measurements of Bone Remodeling.....	64
5.3.4. Intra-specimen and Inter-observer Variation.....	69
5.4. Results.....	72
5.4.1. Image Processing and Validation.....	72
5.4.2. Region of Interest Size.....	74
5.4.3. Alterations in Three-Dimensional Measures of Bone Resorption and Formation Associated with Established Ovariectomy.....	74
5.5. Discussion.....	78

5.6. Acknowledgements	83
6. Microscopic Tissue Damage Forms Preferentially Near Remodeling Cavities in Vertebral Cancellous Bone	84
6.1. Abstract.....	84
6.2. Introduction.....	85
6.3. Methods.....	89
6.3.1. Sample Preparation and Mechanical Testing	89
6.3.2. Image Acquisition and Processing	90
6.3.3. Spatial Association of Microdamage and Resorption Cavities	93
6.3.4. Statistical Analyses	94
6.4. Results	95
6.5. Discussion	102
6.6. Acknowledgements	107
7. Conclusions and Future Directions	108
Appendix	117
A.1 Voxel size and measures of individual resorption cavities in three-dimensional images of cancellous bone.....	118
A.1.1 Abstract	118
A.1.2 Introduction	119
A.1.3 Methods.....	121
A.1.4 Results	125
A.1.5 Discussion	130
A.1.6 Acknowledgements	135
A.2 Code	136
References	315

LIST OF FIGURES

Figure 1-1. Postmenopausal osteoporosis results in increased bone turnover. Increased bone turnover is associated with increased risk of fracture. In vitro studies have determined the effects of estrogen depletion on bone cell proliferation, differentiation, and apoptosis. However, it is not known how changes in bone cell biology influence bone remodeling in vivo. It is also unclear precisely how increased bone turnover increases bone fragility and leads to increased fracture risk.	3
Figure 2-1. (Left) A single trabecula is shown with a resorption cavity. (Right) Eventually resorption may lead to complete disconnection of the trabecula. Used with kind permission from Springer Science+Business Media: Zeitschrift fur Rheumatologie, Age-related changes in bone mass, structure, and strength – effects of loading, 59(7) 2000, pp. 5-6, L. Mosekilde, Figs 4, 7, © Steinkopff Verlag 2000.....	11
Figure 2-2. Osteoclast differentiation requires that M-CSF binds to osteoclast progenitor cell's receptor c-fms. When the membrane bound RANKL binds to the RANK receptor on the promonocyte of the osteoclast cell lineage, the promonocyte differentiates to become a multinucleated mature osteoclast.....	12
Figure 2-3. A complex feedback system of extracellular signaling factors exists that regulates local bone remodeling activity. Promotion (+) and inhibition (-) of bone cell activity occurs both by secretion of soluble factors and through cell-cell contact of membrane bound ligands and receptors. Cells of the hematopoietic stem cell (HSC) lineage are responsible for bone resorption. Cells of the stromal mesenchymal stem cell (MSC) lineage eventually differentiate into osteoblasts, which form new bone. Osteoblasts may then either apoptose or differentiate into mononuclear cells or osteocytes. Inspired by [1].....	16
Figure 3-1. In traditional histomorphometry, analysis of histological sections is performed using a gridded eyepiece reticule. A cross section of bone tissue under a UV light source is shown as it would appear under a gridded eyepiece. Three-dimensional measures of area are approximated from the number of gridline–tissue intersections (x). Measures of volume are approximated from the number of gridpoint – tissue intersections (circles).....	22
Figure 3-2. Double formation labeling is applied to mark where bone formation is occurring at multiple time points. (A) Osteoblasts secrete osteoid on the bone surface (brown). (B) A fluorochrome is injected (orange) in vivo. The fluorochrome binds to calcium in the osteoid recently secreted by osteoblasts. (C) As bone continues to form, the fluorochrome is sealed beneath the bone surface. Three to ten days later, a second fluorochrome (green) is applied. (D) After one day, the second fluorochrome is buried beneath the bone surface.....	27
Figure 4-1. Serial fluorescence microscopy is accomplished using a CNC milling device. An attached grayscale camera, automated filterset changer, and microscope objective lens are used to serially and automatically acquire images of each cross-section.....	38
Figure 4-2. An image of a bone specimen (bright) is shown. (Left) The original image shows reduced signal intensity at the periphery of the FOV. (Right) The same image after correction for non-uniform illumination.....	41

Figure 4-3. The original image of trabecular bone shows considerable sub-surface signal observed as blurring (Arrows). The underlying image is convolved, scaled, and then subtracted from the original image to adjust for sub-surface signal. The adjusted image does not have the blurred regions and the edges of the bone are more defined.	44
Figure 4-4. Images are segmented using an iterative thresholding algorithm adapted from Wu, et al. (2001). An original image is segmented using a matrix of pixel-specific threshold values. Subsequently, the threshold values are adjusted and the original image is segmented again. This process is repeated for a number of iterations selected by user comparison to the gray-scale image.....	46
Figure 4-5. An original image shows bone (bright) and surrounding marrow (dark). More bone and less marrow is included in the segmentation as the algorithm iterates (bottom left). After the final (25th) iteration, some free floating pixels representing marrow components (noise) are removed (bottom right). Note the improvement in the segmentation in comparison to global thresholding (top right).	48
Figure 4-6. Images of cancellous bone acquired through serial block face imaging processed with previously described techniques (left) and with the current techniques (right). Both images have been visualized using identical isosurface smoothing parameters. Images processed using the current technique show a much smoother bone surface that is similar to images obtained using scanning electron microscopy (Mosekilde, 1990).	49
Figure 4-7. (Left) A gray scale image of the cancellous bone surface collected using serial block face imaging is shown. An eroded surface is indicated by bone surface irregularities (arrows). Non-eroded surfaces are smooth (lower right corner of the image). (Right) A fully processed version of the same image in which the bone surface irregularities characteristic of eroded surface can also be seen.....	49
Figure 4-8. An isosurface rendering of a location on the bone surface is shown illustrating surface texture variations characteristic of eroded surfaces.	50
Figure 4-9. An image of a rat tibial metaphysis, collected using serial milling and processed using the methods described in the current study, is shown. Two different fluorescent markers were applied to the animal prior to euthanasia, xylenol orange (shown in orange) followed one week later by calcein (shown in green). Three-dimensional images of the bone and the fluorescent markers of bone formation were collected using serial block face imaging with fluorescent markers processed using global thresholding. Future work will refine and validate the image processing techniques used to examine the fluorescent markers to allow direct, three-dimensional measures of eroded surfaces as well as the fluorescent markers of bone formation.....	54
Figure 5-1. (A) Image processing is validated through direct comparison of pixels in the traced labels to corresponding pixels in automatically processed images. (B) A double labeled formation event, consisting of a pair of spatially correlated formation labels (xylenol orange, calcein green), is visible by making the bone surface partially transparent. Images of bone tissue (off-white) and fluorescent bone formation labels (xylenol orange, calcein green) are shown for (C) ovariectomized (OVX) and (D) sham operated animals.....	63

Figure 5-2. (A) Resorption cavities are identified as indentations on the bone surface. The horizontal lines that appear on the reconstructed bone surface correspond to cross-sections during image acquisition. (B) An observer manually traces the resorption cavity (white). (C,D) Maximum Cavity depth is determined by approximating the ‘pre-resorbed’ bone surface above a resorption cavity using a three-dimensional spline (see Methods for a description of the curve fitting). (E) A cross-section of a resorption cavity with the ‘pre-resorbed’ surface applied is shown. Cavity depth was measured as the maximum distance between the ‘pre-resorbed’ surface and the eroded surface (dashed line). 66

Figure 5-3. A double labeled formation event is identified as a pair of spatially associated formation labels. For quantification, the fluorescent labels are flattened. Mineral apposition rate is calculated as the volume between the flattened labels divided by the label surface area and the inter-label time. 68

Figure 5-4. Mineral apposition rate in ovariectomized animals (each line is an individual rat) measured in three different regions of interest (caudal, central+caudal, and cranial+central+caudal) are shown illustrating that measurements made in 2 mm³ regions of interest are similar to those achieved using larger regions of interest. 70

Figure 5-5. The effect of region of interest size on 3D histomorphometry measurements is illustrated. Each line connects data from the same individual. Bone resorption and formation measures achieved in ovariectomized rats are shown for caudal (1 mm³), central+caudal (2 mm³), and cranial+central+caudal (3 mm³) regions of interest. Measurement values converged and variability decreased in larger regions of interest. 71

Figure 5-6. Histograms of cavity surface area, maximum cavity depth, cavity volume, and double labeled formation event size are shown. The histograms represent pooled data from all of the specimens examined in the study..... 76

Figure 6-1. (A) A trabecula is shown with xylenol orange (staining pre-existing microdamage) and calcein green (staining both pre-existing microdamage and microdamage caused by loading) overlays. (B) Stain that did not extend more than 5 μ m into the bone surface was classified as non-specific surface staining and was removed (arrows). (C) Sites with both xylenol orange and calcein were removed, leaving only microdamage generated by controlled mechanical loading. 92

Figure 6-2. (A) The regions of bone with the largest microdamage sites were identified (32 per specimen). Subsequently, regions of bone without microdamage were selected at random (32 per specimen). (B) The bone surface around each region was displayed without showing microdamage so that the observer was blinded to the presence of microdamage. (C) The observer then determined if a resorption cavity was present in the image by observing an indentation on the bone surface associated with eroded surface (arrows). 94

Figure 6-3. (A) The histogram of volume per microdamage site was positively skewed. (B) Microdamage surface to volume ratio was normally distributed..... 96

Figure 6-4. (A) The distribution of microdamage generated by controlled mechanical testing (green) within cancellous bone (off-white) is shown in one specimen. (B) Microdamage (arrows) was often diffuse, but tended to be crack-like in that it was thin and broad. 96

Figure 6-5. The proportion of bone surfaces with cavities was greater in sites with microdamage than in randomly identified locations without microdamage. Locations with microdamage had a relative risk of 2.39 to have a neighboring cavity.....	98
Figure 6-6. Microscopic tissue damage generated by controlled mechanical loading (green) was more likely to be near resorption cavities.	99
Figure 6-7. Damage volume per bone volume (DV/BV) was positively correlated with percent eroded surface (ES/BS).	100
Figure 7-1. The techniques presented in this dissertation can be applied to study how metabolic bone disease influence bone turnover. The techniques can also be applied to determine the mechanisms behind how changes in bone turnover influence bone fragility and fracture risk. ..	112
Figure 7-2. Changes in bone cell function determine the number and size of individual remodeling events. Changes in bone cell function also influence recruitment and differentiation of other bone cells.	113
Figure A-1. (A) Initial identification of a resorption cavity on the cancellous bone surface is performed by observing an indentation on the cancellous bone surface in the three-dimensional image (arrows). (B) The presence of a cavity is then confirmed by viewing a two dimensional cross-section of the gray-scale region where the irregular bone surface (arrows). (C) The cavity is then labeled in the 3D image and recorded for future analysis.	124
Figure A-2. A region of cancellous bone with a resorption cavity is shown at each of the image resolutions considered in the study. The three-dimensional image is shown on the left and a single cross-section from the image is shown on the right. As the voxel size increases the resorption cavity becomes increasingly difficult to observe and bone surface irregularities characteristic of resorption cavities become more difficult to distinguish from the bone surfaces. In images with the largest voxel size the eroded surfaces were not discernible from other bone surfaces and no cavities were found.	128
Figure A-3. The relationship between average number of cavities observed in each specimen and the in-plane resolution of the images is shown (out-of-plane resolution is 5 mm/voxel except for the largest images where it is 10mm/voxel). There was an exponential decline in the number of cavities observed as voxel size increased. Error bars represent standard deviations.	129
Figure A-4. Histograms showing the distribution of cavity surface size for three different image resolutions (identified by in-plane pixel size). Cavities observed in 0.7 mm and 1.4 mm images showed a similar size distribution, while cavities observed in lower resolution images (2.8 mm shown here) were less numerous and were larger in surface size.	130

LIST OF TABLES

Table 3-1. Primary histomorphometry measures are listed below along with standardized two-dimensional and three-dimensional abbreviations. © (1987) Wiley. Used with permission from AM Parfitt; MK Drezner; FH Glorieux; JA Kanis; H Malluche; PJ Meunier; SM Ott; RR Recker. “Bone histomorphometry: Standardization of nomenclature, symbols, and units: Report of the asbmr histomorphometry nomenclature committee” J Bone Miner Res. Mary Ann Liebert, Inc. 24	
Table 3-2. Derived static (structural) and dynamic (kinetic) histomorphometry measures. © (1987) Wiley. Used with permission from AM Parfitt; MK Drezner; FH Glorieux; JA Kanis; H Malluche; PJ Meunier; SM Ott; RR Recker. “Bone histomorphometry: Standardization of nomenclature, symbols, and units: Report of the asbmr histomorphometry nomenclature committee” J Bone Miner Res. Mary Ann Liebert, Inc.	28
Table 4-1. Bone volume fraction, bone surface density, and the mean surface curvature of cancellous bone isosurfaces (mean \pm SD) were measured in 5 rat lumbar vertebrae using the previous image processing techniques and the revised techniques presented in the current study. Non uniform illumination (NUI) correction was applied in all cases.	51
Table 5-1. Measurements achieved in ovariectomized animals (n=5) using traditional two-dimensional techniques are shown along with three-dimensional measures from the same specimens (mean \pm SD). The paired difference between the two approaches is determined within each specimen and the average value across all specimens is shown.	73
Table 5-2. Static histomorphometry measures made using the three-dimensional approach are shown (mean \pm SD, n=5 per group).....	77
Table 5-3. Measures of bone formation made using the 3D approach are shown (mean \pm SD, n=5 per group).....	77
Table 6-1. Static histomorphometry parameters and microdamage measured in three-dimensional images as well as apparent biomechanical properties are shown.	97
Table 6-2. Correlation coefficients between static histomorphometry and microdamage measures (95% confidence intervals). Correlations that are significant do not include zero in the 95% confidence interval.....	101
Table 6-3. Correlation coefficients between mechanical properties and microdamage measures are listed with 95% confidence intervals in parentheses. No correlations were significant.	101
Table A-1. Measurements made from three-dimensional images of rat vertebral cancellous bone were performed at five different image voxel sizes (Mean \pm SD, n=5 specimens). Results with the same superscript letter are not significantly different from one another (p > 0.05, paired t-test with Bonferroni multiple comparisons). No cavities were observed in images with the largest voxel size.	131

1. Introduction

1.1 Motivation

Osteoporosis is a disease characterized by low bone mass and increased fracture risk. Millions of osteoporosis related fractures occur each year. Medical treatment of osteoporosis related fractures costs more than \$18 billion annually [2]. The costs associated with osteoporosis are expected to increase as a result of extended life expectancy and an expanding population [3]. Improvements in our understanding of the development and treatment of osteoporosis may help reduce the incidence of osteoporosis related fractures and improve the efficacy of clinical therapeutic interventions.

Osteoporosis is diagnosed clinically using the World Health Organization FRAX tool. The FRAX tool uses risk factors as inputs to a statistical computer model that determines a person's 10 year risk of osteoporosis related fracture. Risk factors include sex, body mass index, nationality, previous fragility fracture, family history of fractures, tobacco use, and areal bone mineral density (aBMD). Areal bone mineral density is a major factor determining FRAX score [4, 5]. Areal bone mineral density is determined clinically using dual x-ray absorptiometry (DXA). Dual x-ray absorptiometry uses x-rays with two energy levels to provide information about the bone mineral content present. Areal bone mineral density is defined as the mineral content normalized by the cross-sectional area of the region of the bone being considered (g/cm^2). Areal BMD explains less than half of the variability in fracture risk in both men and women during anti-resorptive osteoporosis drug therapy [6-9].

Bone turnover may be important in determining bone fragility and fracture risk. Bone turnover occurs through the process of bone remodeling. In bone remodeling, osteoclasts remove bone leaving a resorption cavity. Subsequently, osteoblasts form new bone within the resorption cavity [10]. Bone remodeling is performed by discrete groupings of cells called basic multicellular units (BMU). Bone remodeling is the primary process of modifying bone mass in adults and is therefore believed to play an important role in determining bone loss and subsequent fracture risk in states of metabolic bone disease. Biomarkers of bone remodeling in the body have been correlated with fracture risk independent of bone mineral density [11]. However, precisely how bone remodeling influences bone biomechanics and fracture risk is currently unknown [12]. Bone fragility and fracture risk may be sensitive to changes in the size and number of individual remodeling events [7]. Changes in the number and size of individual remodeling events ultimately determine the amount of bone turnover in the body.

Changes in bone turnover are characteristic of many metabolic bone diseases. The changes in bone turnover are influenced by changes in bone biology. Bone biology regulates bone cell functions including proliferation, differentiation, vigor, motility, and apoptosis. Bone cell functions, in turn, determine the number and size of individual remodeling events. Furthermore, how changes in bone turnover influence fracture risk is not known. Several studies suggest that the number, size, and location of remodeling events can influence cancellous bone biomechanics [13-15]. Understanding the link between bone turnover and increased fracture risk requires an understanding of how changes in individual remodeling events influence cancellous bone biomechanics (Figure 1-1).

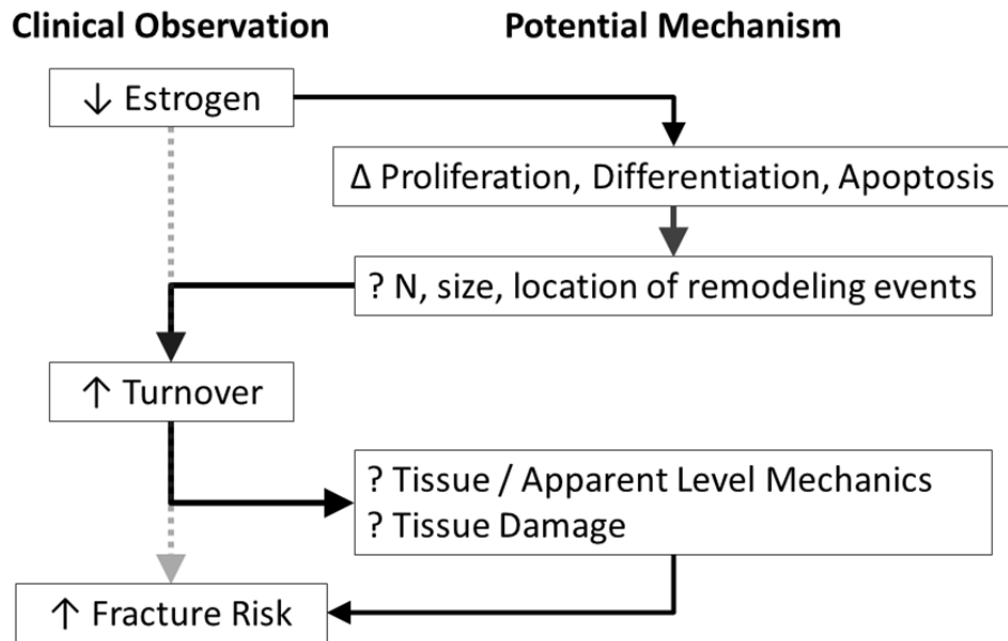


Figure 1-1. Metabolic bone disease can influence fracture risk by altering bone turnover. Alterations in bone biology in metabolic bone disease states influence bone cell functions. Changes in bone cell function regulate individual remodeling events and determine the rates of bone turnover in the body. Bone turnover may impair cancellous bone biomechanics, leading to increased bone fragility and fracture risk.

The overall goal of this line of investigation is to understand how bone biology influences bone biomechanics and fragility. The specific aims of this dissertation are to (1) develop techniques to acquire three-dimensional images of bone remodeling events in whole cancellous bone specimens, (2) determine how estrogen depletion influences individual remodeling events using a common animal model of post-menopausal osteoporosis, and (3) determine how bone remodeling influences cancellous bone biomechanics and damage formation (Figure 1-1).

1.2 Outline

This dissertation describes techniques to determine how bone remodeling is influenced by changes in bone biology. A study was also performed to determine how bone remodeling influences local and continuum level cancellous bone biomechanics. Chapter 2 provides background information on the structure and function of bone, bone remodeling, and bone biomechanics. An overview of the pathogenesis of bone remodeling is provided in the context of regulatory factors involved in bone remodeling. Chapter 3 describes standardized techniques used to quantify bone microarchitecture and bone remodeling. The techniques' strengths and limitations are also discussed. Chapter 4 describes a serial milling based technique to image bone remodeling throughout cancellous bone specimens. The technique provides the capability to quantify the relationship between bone cell biology and bone remodeling activity in vivo. The information in Chapter 4 was published in the Journal of Microscopy. Chapter 5 describes how estrogen depletion affects individual bone remodeling events in vivo. The information in Chapter 5 was published in the Journal of Bone and Mineral Research. Chapter 6 quantifies the relationships between bone remodeling, microscopic tissue damage, and cancellous bone biomechanics. The results of this investigation provide direct evidence that bone remodeling influences failure processes in cancellous bone. This work was recently submitted for publication in the Bone. Portions of chapters 4 through 6 were presented at annual meetings of the Orthopaedic Research Society, International Society of Bone Morphometry, American Society for Bone and Mineral Research, Biomedical Engineering Society, and American Society of Mechanical

Engineering. Finally, chapter 7 discusses the implications of the findings in this dissertation and a framework for future investigations is provided.

2 Bone

2.1 Bone Function and Structure

The human skeleton is composed of 206 bones. Bones protect vital organs such as the brain, lungs, and heart. Bones also provide support for the body's soft tissues and allow for locomotion. The skeleton is the main site of hematopoiesis in the body (the formation of cells related to the blood). Bones store minerals for vital functions.

Bone is a complex polymer-ceramic composite material. Bone is comprised of an inorganic phase of mineral and an organic phase consisting of organic matrix, proteins, cells and water. The inorganic phase of bone is comprised largely of hydroxyapatite. Minerals such as potassium, magnesium, fluoride, and others exist as impurities within individual hydroxyapatite crystals [16]. The organic phase of bone consists of approximately 90% collagen. The remaining 10% consists of growth factors, cytokines, proteoglycans, extra cellular matrix proteins, etc. [16, 17].

The structure of bone is hierarchical in nature and has distinct structures at multiple characteristic length scales. At the smallest length scale, collagen molecules are bound together on end by hydroxyapatite crystals. Repeating units of collagen and hydroxyapatite form 100nm long collagen fibrils. Groupings of parallel collagen fibrils form a network of parallel collagen fibers of about 10 μm in size. Different arrangement of the collagen fibers results in different forms of bone. Lamellar bone is laid down in a highly ordered pattern of parallel sheets. Woven bone, on the other hand, is laid down in

a quasi-random pattern and is associated with rapid bone formation such as that in fracture healing.

Individual lamellar sheets are laid down in alternating orientations to form concentric rings known as osteons. Osteons are constructed from concentric lamellar sheets. At the center of an osteon is a hollow channel known as Haversian canal. Haversian canals provide a space for blood vessels and nerves. Channels perpendicular to the primary axis of osteons are referred to as Volkmann's canals. Volkmann's canals also act as spaces for blood vessels and nerves.

At the millimeter scale, multiple bone structural units constitute what is known as cortical or cancellous bone. Cortical bone forms a shell around all bones. Cortical bone is found in long bones such as the femur, tibia, and humerus. Cortical bone is very dense, with a porosity of 5-10%. Cancellous bone has a sparse structure of 60% - 90% porosity [18, 19]. Cancellous bone consists of a network of individual trabecular struts. The trabeculae may either appear rod-like or plate-like. Trabeculae are approximately 100-200 μm in thickness and up to 1 mm in length. Trabeculae tend to be oriented along lines of apparent principal stress and strain [20, 21]. Cancellous bone is constructed from bone structural units of parallel lamellae that are formed along the axis of individual trabecular struts. Cancellous bone is found primarily at the metaphyseal ends of long bones and in the vertebrae, femora, pelvis, and the skull.

2.2 Cancellous Bone Mechanics

The mechanical properties of bone are influenced by tissue level properties such as mineralization and collagen fiber cross-linking [22, 23]. Cancellous bone is also influenced by its apparent level properties such as porosity, apparent density, trabecular microarchitecture, and anatomical site [24]. Apparent level mechanical properties are those determined in specimens of sufficient physical dimension that they act as a continuum (greater than 5 mm) [25]. The microarchitecture of the trabecular network gives cancellous bone orthotropic material symmetry at the apparent level. However, for simplicity, cancellous bone material properties are often modeled as transversely isotropic [21, 26-31]. Typical ranges of mechanical properties are listed here with respect to the direction of primary trabecular orientation.

The apparent elastic modulus of human cancellous bone ranges from 67-1100 MPa [32-34]. The wide range in elastic modulus is due in part to the fact that elastic modulus varies by a power law relationship with apparent density (typically expressed as g/cm^3) [35, 36]. Additionally, the stress-strain curve for cancellous bone is non-linear. The non-linearity makes estimating elastic modulus in a repeatable manner difficult. A specialized technique has been proposed by Morgan, et al. [37] that substantially reduces variability in elastic modulus estimation. The technique estimates elastic modulus as the initial slope of a second-order polynomial fit to the elastic regime (0-0.2% strain) of the stress strain curve. The technique can then be applied to determine a specimen's yield point using the 0.2% offset strain method [34].

The strength of cancellous bone depends on microarchitectural attributes, mineralization, and age. Ultimate stress in compression varies from 2 MPa in vertebrae to over 7 MPa in the femur [33, 38, 39]. Strength depends on strain rate, apparent density, and loading direction [35, 36]. According to cross-sectional data, human bone strength peaks at age 40 and decreases by approximately 10% per decade in the vertebrae [40, 41]. Because cancellous bone microarchitecture is anisotropic, strength can be up to 10 times higher in the longitudinal direction than in the transverse direction [42]. Strength is also greater in compression than in tension. The difference in compressive and tensile strength has been observed to vary linearly with modulus [43]. However, this relationship has, so far, only been observed in bovine cancellous bone.

2.2.1 Damage in Cancellous Bone

An interesting property of cancellous bone is that its yield strain does not depend on density or microarchitectural parameters. Cancellous bone yields in compression at a strain of 0.8% [32]. The ultimate strain of cancellous bone varies only with loading direction. Cancellous bone ultimate strain is 1.45% in compression and 1.59% in tension [32].

Damage in cancellous bone is observed mechanically as reductions in modulus or histologically as microscopic cracks. Reductions in the modulus (i.e. the amount of damage) have been correlated with residual strain [44], indicating that damage is related to plastic deformation in cancellous bone. Reductions in modulus have been observed in

cancellous bone specimens subjected to loading past yield and cyclic loading pre-yield [44, 45]. Damage is also evaluated histologically.

The standard method of observing and quantifying microscopic tissue damage in bone specimens is through examination of histologic sections (see chapter 3). Multiple forms of microscopic tissue damage have been observed including linear microcracks, diffuse damage (diffuse sub-micron scale cracking) and trabecular microfractures [46, 47]. The forms of microscopic tissue damage are collectively referred to as microdamage. Microdamage is thought to influence bone fragility and osteoporosis related fracture risk [48, 49]. However, the relationship between mechanical damage in cancellous bone and microscopic tissue damage is currently not known.

2.3 Osteoclasts and Bone Resorption

The main function of osteoclasts is to resorb old and damaged bone [50, 51]. Osteoclasts are giant multinucleated cells. The mature osteoclast is approximately 20 to 100 μm in diameter and can contain as many as 50 nuclei. Active mature osteoclasts are polarized, with a ruffled border. The ruffled border features densely packed vitronectin ($\alpha_v\beta_3$ integrin) receptors that allow the cell to attach to the bone surface [52]. The integrins ensure a controlled microenvironment beneath the cell necessary for bone resorption to occur. The ruffled border of the osteoclast secretes protons and cathepsin-k [53-55]. Together, these compounds degrade the inorganic bone matrix and matrix bound proteins (e.g. collagen, growth factors, and bone morphogenic proteins) [56]. As the osteoclast resorbs bone, it may also travel along the bone surface. Resorption by

osteoclasts leaves a resorption cavity with an irregular scalloped surface, known as a Howship's lacuna (Figure 2-1).

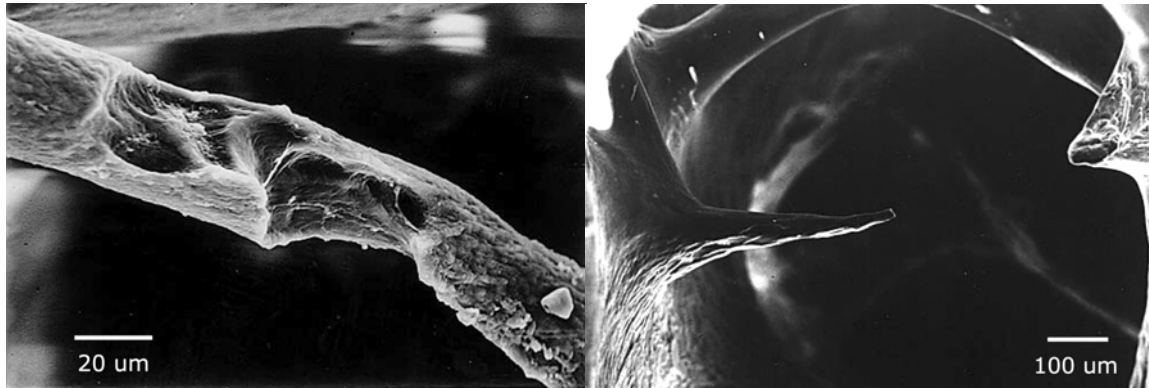


Figure 2-1 (Left) A single trabecula is shown with a resorption cavity. (Right) Eventually resorption may lead to complete disconnection of the trabecula. Used with kind permission from Springer Science+Business Media: *Zeitschrift für Rheumatologie*, Age-related changes in bone mass, structure, and strength – effects of loading, 59(7) 2000, pp. 5-6, L. Mosekilde, Figs 4, 7, © Steinkopff Verlag 2000.

The osteoclast cell type is derived from a hematopoietic stem cell (HSC). The first step in osteoclast differentiation requires that M-CSF (Macrophage colony stimulating factor) binds to the c-fms receptor of the uncommitted osteoclast precursor (Figure 2-2a). The CFU-GM (colony forming unit – granulocyte-macrophage progenitor) then proliferates and differentiates into a promonocyte [55, 57]. The promonocyte will express RANK (receptor activator of nuclear factor kappa-B). The next step in osteoclast differentiation involves cell-cell contact with a pre-osteoblast or osteoblast. RANKL (RANK ligand) is a membrane bound ligand expressed by osteoblasts and marrow stromal cells. When RANKL binds to the RANK receptor of the promonocyte, the promonocyte differentiates into a preosteoclast (Figure 2-2b). The preosteoclast then becomes multinucleated by fusing with other cells. Fusion and differentiation of preosteoclasts to a late preosteoclast is upregulated due to osteoblast and marrow stromal

cell RANKL expression [58-61]. The late preosteoclast lacks proliferative potential and may only differentiate further or undergo a process of programmed cell death known as apoptosis. The late preosteoclast expresses calcitonin receptors and $\alpha_v\beta_3$ integrins. The $\alpha_v\beta_3$ integrins allow the osteoclast to bind to matrix proteins [52, 62]. Mature osteoclasts are activated when exposed to collagenase secreted by bone lining cells and when $\alpha_v\beta_3$ integrins bind to bone matrix proteins [62, 63]. Mature osteoclasts resorb bone for approximately one week. Osteoclasts then migrate to the nearby marrow space to undergo a process of programmed cell death called apoptosis. The causes behind activation of the pathways leading to osteoclast apoptosis are not well understood. Apoptosis is believed to increase in response to estrogen and TGF- β (transforming growth factor β) [64] as well as a number of other factors (see 2.6).

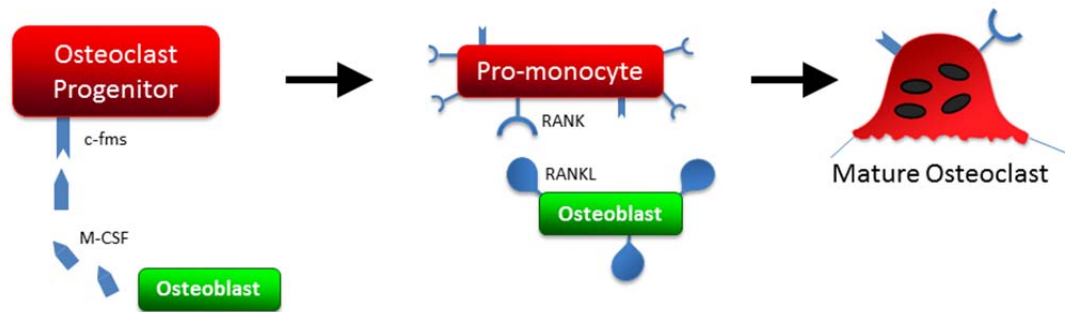


Figure 2-2. Osteoclast differentiation requires that M-CSF bind to osteoclast progenitor cell's receptor c-fms. When the membrane bound RANKL binds to the RANK receptor on the promonocyte of the osteoclast cell lineage, the promonocyte differentiates to become a multinucleated mature osteoclast.

2.4 Osteoblasts and Bone Matrix Production

Osteoblasts are the cells responsible for forming bone. Osteoblasts are derived from pluripotent mesenchymal stem cells (MSC) within the marrow space. The fully differentiated mature osteoblast is a mononuclear cuboidal cell of approximately $700 \mu\text{m}^3$

in volume [65]. Mature active osteoblasts form new bone by secreting osteoid. Osteoblasts produce a layer of osteoid between 0.1 and 1 μm in thickness per day [66, 67]. Osteoid is comprised of 90% collagen and 10% noncollagenous proteins. The noncollagenous proteins include growth factors, bone morphogenic proteins, and many other compounds [68]. Collagen forms a structure that aids in the osteoid mineralization process. Collagen fibers are laid down in a parallel fashion with the last 25% of each fiber being staggered. After one week, hydroxyapatite begins to form at the gap locations between fibers [69]. Hydroxyapatite provides rigidity to the collagen network. Individual collagen fibers are cross-linked by pyridinoline [70]. The cross-links stabilize the collagen network [19]. A highly ordered collagen network is required for lamellar bone to form. In fracture healing and in early grown in humans, bone is formed too rapidly for an ordered collagen network to be formed. For this reason, woven bone is associated with rapid bone formation [19, 71].

When an osteoblast ceases to form bone, it may either apoptose or differentiate into an osteocyte or a bone lining cell [72]. Inactive osteoblasts develop a flat appearance and are referred to as bone lining cells. Alternatively, osteoblasts may become covered by osteoid and eventually become osteocytes. Osteocytes reside within lacunae within the bone matrix. Osteocytes communicate with other osteocytes through dendritic structures inside small channels. The small channels pass through the bone matrix and are called canaliculi. The communication network is thought to be the mechanism by which osteocytes to sense and transduce tissue stresses [73]. Osteocytes may coordinate tissue damage repair by controlling the activity of osteoclasts and osteoblasts [74-77]. During normal activity, bone sustains microscopic tissue damage [78]. The microscopic tissue

damage elicits a repair response. The repair process involves the formation of a microcallus or the initiation of a remodeling event [50, 51, 79]. While a relationship between remodeling and microscopic tissue damage has been observed in cortical bone and in vitro, it has not yet been observed in cancellous bone.

2.5 Bone Remodeling

Bone is a unique material in that it has the ability to repair damage and adapt to changes in loading [50, 80, 81]. Bone remodeling is also the main method of modifying bone structure and bone mass in adults. Bone remodeling occurs by the coordinated activities of osteoclasts and osteoblasts. In bone remodeling, osteoclasts resorb old and damaged bone, leaving a resorption cavity (Figure 2-1). Subsequently, the remodeling event enters the ‘reversal’ phase. The reversal phase involves local cell signaling that causes active osteoclasts to apoptose and osteoblast precursors to differentiate and attach to the bone surface. Approximately one week later, osteoblasts begin to secrete osteoid within the resorption cavity [82]. The osteoid mineralizes over one week to become hard bone tissue. The formation period of bone remodeling is substantially longer than the resorption period. The resorption period occurs over approximately one week, whereas the formation period lasts on the order of months [83].

Bone remodeling occurs by teams of cells called a basic multicellular unit (BMU). BMUs are initiated at discrete locations on the bone surface [84, 85]. The birthrate, number, and size of BMUs determine the overall rate of bone turnover in the body [86]. The rate of bone turnover is important because it influences bone volume

fraction and tissue degree of mineralization and may also result in trabecular thinning and fenestration [87].

The amount of bone turnover in the body has been associated with increased fracture risk [88]. Bone remodeling occurs as focal events on the bone surface, suggesting that remodeling is regulated not only by circulating regulatory factors, but also by local mechanisms in the bone microenvironment. The rates of proliferation, differentiation, activity, and apoptosis of cells from the osteoclast and osteoblast lineages determine BMU size, number, longevity, and progression across the bone surface (Figure 2-3). The mechanisms that regulate remodeling events are therefore important in understanding how bone turnover influences bone fragility and fracture risk. Knowledge of the biological factors that regulate bone remodeling has been gained from in vitro studies and is discussed below.

2.6 Regulatory Factors Involved in Bone Remodeling

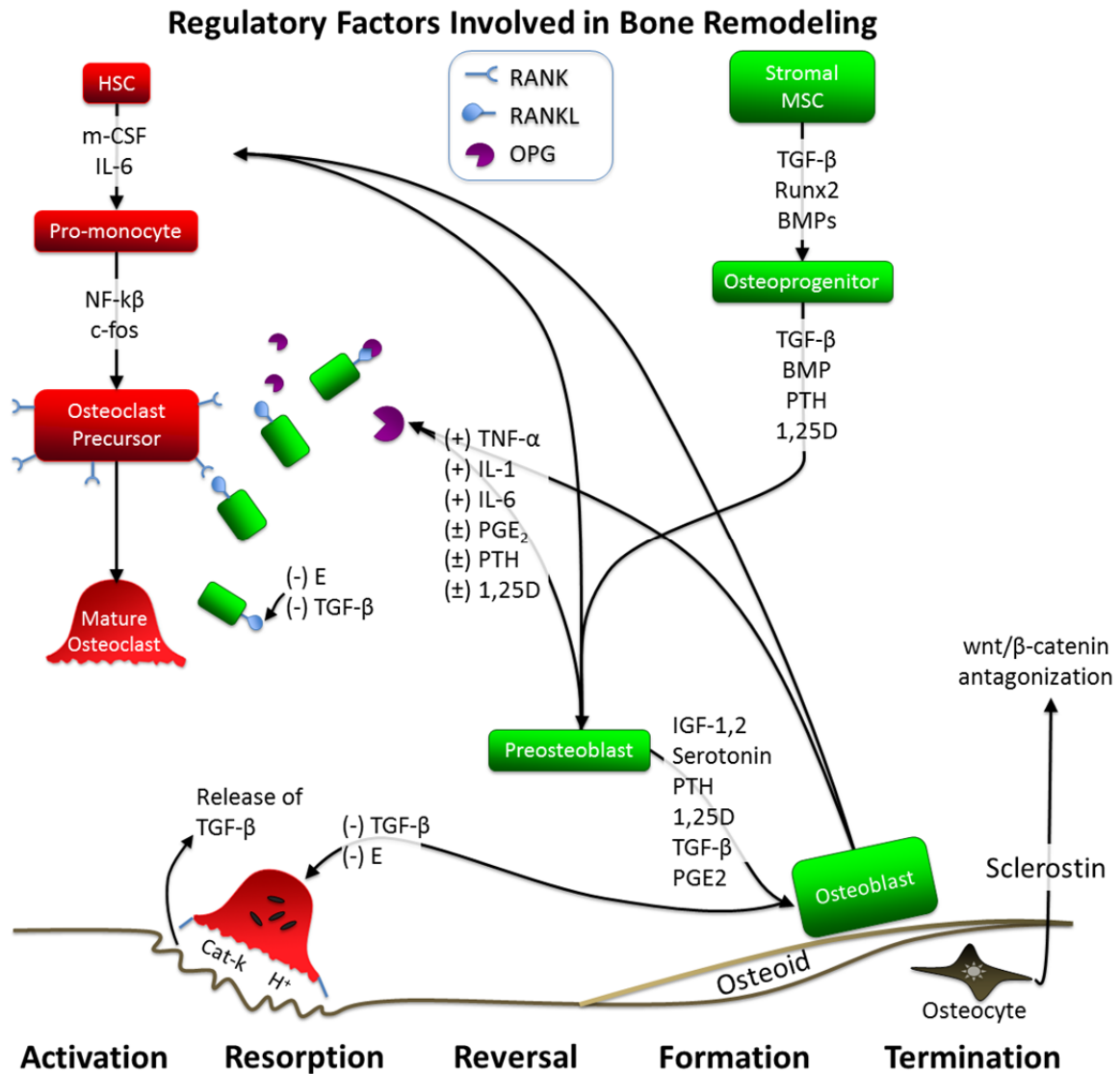


Figure 2-3. A complex feedback system of extracellular signaling factors exists that regulates local bone remodeling activity. Promotion (+) and inhibition (-) of bone cell activity occurs both by secretion of soluble factors and through cell-cell contact of membrane bound ligands and receptors. Cells of the hematopoietic stem cell (HSC) lineage are responsible for bone resorption. Cells of the stromal mesenchymal stem cell (MSC) lineage eventually differentiate into osteoblasts, which form new bone. Osteoblasts may then either apoptose or differentiate into mononuclear cells or osteocytes. Inspired by [1].

In addition to bone formation, osteoblasts are also partly responsible for regulating osteoclast activity. RANKL is a membrane bound factor on osteoblasts, pre-osteoblasts, and stromal cells. RANKL is central to the regulation of osteoclast activity. RANKL expression is upregulated by a number of factors including PTH (parathyroid hormone), IL-1 (interleukin-1), PGE₂ (prostaglandin E₂), PTH rp (parathyroid hormone related protein), TNF- α (tumor necrosis factor α), and vitamin D [58, 89-99]. Osteoclast differentiation is negatively regulated by osteoprotegrin (OPG). OPG is a decoy receptor for RANKL. OPG inhibits osteoclast differentiation by binding to RANK [100, 101]. As RANKL is membrane bound, its supply in the bone microenvironment is limited. The ratio of OPG to RANKL therefore determines the rate at which osteoclasts are able to differentiate. Osteoclastogenesis is positively, yet indirectly, regulated by TNF- α . Osteoclastogenesis is negatively regulated by TGF- β (transforming growth factor β), but positively by TNF- α [102, 103]. As osteoclasts resorb bone, they release matrix bound proteins including TGF- β . The more resorption that occurs, the more this inhibitory factor is released into the bone microenvironment. In this way, osteoclasts exhibit a self-regulatory mechanism that limits resorption [104, 105]. Osteoclastogenesis is also negatively regulated via the wnt/ β -catenin pathway. The wnt/ β -catenin pathway can act indirectly to increase the ratio of OPG to RANKL [106, 107]. However, this effect is mediated by the expression of sclerostin. Sclerostin inhibits activation of the wnt/ β -catenin pathway and, therefore, osteoblastogenesis and bone formation [108, 109].

Osteoblast function is regulated by a number of factors, including BMPs, TGF- β , IGF-1 and -2, vitamin D, PGE₂, and PTH [110, 111]. Bone morphogenic proteins induce transcription factors which trigger differentiation of the osteoblast progenitor cell [112-

114]. Runx2 is a necessary transcription factor involved in osteoblast differentiation [115]. Runx2 is regulated by smads to increase osteoblastogenesis [116]. Smads are proteins that act intracellularly to transduce various TGF- β related ligands (e.g. BMPs) to the cell nucleus. Smads therefore enable transcription of ligands from the TGF- β superfamily. TGF- β and BMP-2 are two such factors that upregulate osteoblast differentiation and bone formation [117, 118]. Other factors including PTH and PGE2 inhibit apoptosis of mature osteoblasts [119]. PTH also acts by upregulating IGF-1 expression. The increase in IGF-1 results in an increase in osteoblast precursor proliferation and an increase in collagen production by mature osteoblasts [120].

2.7 Osteoporosis

Osteoporosis is characterized by low bone mass. Reductions in bone mass are caused by bone remodeling. Bone remodeling is influenced by systemic and local regulatory factors. In general, factors that promote bone resorption and suppress bone formation may lead to reduced bone mass and, eventually, osteoporosis. There are two types of osteoporosis: postmenopausal and senile. Postmenopausal (Type I) osteoporosis is caused by estrogen depletion associated with menopause. Senile (Type II) osteoporosis is associated with advanced age. Type I and II osteoporosis fall under the category of primary osteoporosis. Secondary osteoporosis is that caused secondarily by other conditions such as kidney failure, diabetes, hyper- and hypoparathyroidism, cancer, smoking, corticosteroid usage, and many others. Here, I limit my discussion to the more general cases of primary osteoporosis.

Senile osteoporosis is associated with increased levels of PTH as well as reduced vitamin D secretion and calcium absorption. Reduced kidney faculty and/or efficiency in production may cause reductions in vitamin D levels with age [121]. Reduced vitamin D levels may also be a result of deficiencies in diet. The age related insufficiency in circulating vitamin D levels [122] causes decreased calcium absorption in the gut. As a result, PTH secretion is increased. The increased PTH levels increase bone turnover by upregulating factors that promote osteoclast proliferation and differentiation while decreasing osteoclast apoptosis. 1,25 vitamin D deficiency-induced changes in PTH may therefore contribute to age related bone loss by an increase in both the number and longevity of BMUs.

Postmenopausal osteoporosis is associated with estrogen depletion in women. Estrogen limits osteoclast activity while promoting osteoblast activity [88, 123]. The loss of estrogen therefore removes a major inhibitor of osteoclast activity. Estrogen also influences osteoclast and osteoblastic cell activity indirectly through alterations in the production of various local regulatory factors and systemic factors such as PTH, IGF-1, IL-6,7, TNF- α , OPG, TGF- β , BMPs, and FGF. Alterations in each of these factors with estrogen depletion increase M-CSF and RANK/RANKL expression, thereby promoting osteoclast proliferation and differentiation. Alterations in these regulatory factors may explain the increased rate of appearance of new remodeling events observed in estrogen depletion [124]. The same estrogen depletion induced changes in TGF- β and TNF- α concurrently inhibit osteoblast function. Estrogen depletion therefore increases osteoclastic activity while reducing osteoblastic activity. The changes in osteoclast and osteoblast cell functions are observed clinically according to whole body biochemical

markers of bone resorption and formation. However, because no method currently exists to evaluate the effects of changes in these regulatory factors on individual remodeling events in vivo, the biological mechanisms behind bone loss caused by estrogen depletion and age are currently not known.

3. Bone Histomorphometry

Histomorphometry is the standardized method of quantifying microarchitecture, tissue damage, and remodeling within sections of bone. Measures achieved through histomorphometry fall under one of two categories: static and dynamic. Static histomorphometry quantifies morphological and microarchitectural attributes of a bone specimen. Dynamic histomorphometry quantifies time-dependent measures of bone remodeling activity such as the rate of bone formation and the rate of appearance of new remodeling events.

3.1 Static Histomorphometry

Traditional two-dimensional bone histomorphometry is performed by analysis of histology slides. First, a bone specimen is acquired from a human or animal subject. In humans, specimens are often acquired from an iliac crest bone biopsy. The bone specimen is dehydrated using sequential washes of increasingly concentrated alcohol solutions. The dehydrated specimen is then embedded in methyl-methacrylate. Specimens are sectioned to a thickness of 4-7 μm using a microtome. Finally, the sections are mounted on microscope slides for analysis [125].

To perform static histomorphometry, the histology slides are viewed under a microscope with a eyepiece grid reticule using brightfield or fluorescence microscopy [126]. Intersections between the tissue and grid lines points are recorded (Figure 3-1). The number of intersections between the tissue and grid lines is scaled by the grid

spacing to calculate perimeters. Similarly, tissue intersections with grid points are scaled by the area associated with each grid point to determine cross-sectional area [127].

A stereology model is then used to estimate three-dimensional surface area from perimeter measures and volume from cross-sectional area measures. Stereology is a mathematical method used to estimate two-dimensional measures in terms of the three-dimensional morphologies [128]. The most commonly used stereology model in cancellous bone is the parallel plate model. In the parallel plate model, three-dimensional surface areas and volumes are derived from measures of perimeter and cross-sectional area by applying a scaling factor of $4/\pi$ [127]. Most laboratories now use commercially available software packages that reduce labor requirements by allowing a trained observer to digitally trace features of interest. Histomorphometry parameters are then calculated automatically within the software package.

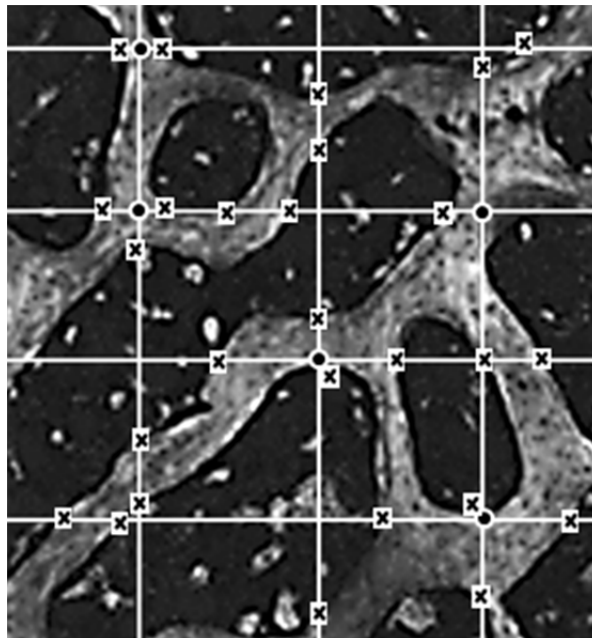


Figure 3-1. In traditional histomorphometry, analysis of histological sections is performed using a eyepiece grid reticule. A cross section of bone tissue under a UV light source is shown as it would appear under a gridded eyepiece. Three-dimensional measures of area are approximated from the number of gridline – tissue intersections (x). Measures of volume are approximated from the number of gridpoint – tissue intersections (circles).

Examples of measures determined using static histomorphometry include bone volume fraction, bone specific surface, trabecular thickness, and trabecular separation. Parameters such as percent eroded surface, the number of osteoblasts, osteoid surface, etc. can also be achieved through static histomorphometry. An extensive list of primary histomorphometry measures and abbreviated notations has been compiled and standardized by the American Society of Bone and Mineral Research (Table 3-1) [129]. Primary histomorphometry measures are those that do not depend on other histomorphometry measures.

Table 3-1. Primary histomorphometry measures are listed below along with standardized two-dimensional and three-dimensional abbreviations. © (1987) Wiley. Used with permission from AM Parfitt; MK Drezner; FH Glorieux; JA Kanis; H Malluche; PJ Meunier; SM Ott; RR Recker. “Bone histomorphometry: Standardization of nomenclature, symbols, and units: Report of the asbmr histomorphometry nomenclature committee” J Bone Miner Res. Mary Ann Liebert, Inc.

Type of Measurement	Name of Measurement	Abbreviations	
		3D	2D
I. Area	Bone volume	BV	B.Ar
	Osteoid volume	OV	O.Ar
	Mineralized volume	Md.V	Md.Ar
	Void volume	Vd.V	Vd.Ar
	Marrow volume	Ma.V	Ma.Ar
	Fibrosis volume	Fb.V	Fb.Ar
	Canal volume	Ca.V	Ca.Ar
	Cell volume	Cc.V	Cc.Ar
	Cytoplasmic volume	Cy.V	Cy.Ar
	Nuclear volume	Nc.V	Nc.Ar
II. Length	Bone interface	BI	B.Bd
	Bone surface	BS	B.Pm
	Osteoid surface	OS	O.Pm
	Eroded surface	ES	E.Pm
	Quiescent surface	QS	Q.Pm
	Mineralized surface	Md.S	Md.Pm
	Osteoblast surface	Ob.S	Ob.Pm
	Single labeled surface	sLS	sL.Pm
	Double labeled surface	dLS	dL.Pm
	Osteoclast surface	Oc.S	Oc.Pm
	Reversal surface	Rv.S	Rv.Pm
III. Distance	Cortical thickness	Ct.Th	Ct.Wi
	Wall thickness	W.Th	W.Wi
	Mineralized thickness	Md.Th	Md.Wi
	Osteoid thickness	O.Th	O.Wi
	Label thickness	L.Th	L.Wi
	Trabecular thickness	Tb.Th	Tb.Wi
	Interstitial thickness	It.Th	It.Wi
	Trabecular diameter	Tb.Dm	-
	Canal radius	Ca.Rd	-
	Cell height	Ce.Ht	-
	Nuclear height	Nc.Ht	-
IV. Number	Osteoblast number	-	N.Ob
	Osteoclast number	-	N.Oc
	Osteocyte number	-	N.Ot
	Nuclear number	-	N.Nc
	Canal number	-	N.Ca
	Seam number	-	N.Sm
	Erosion number	-	NE
	Profile number	-	N.Pf
	Node number	-	N.Nd

Static histomorphometry has also been used to measure microscopic tissue damage in bone specimens [130-132]. Microscopic tissue damage is observed as diffuse damage (submicroscopic cracks), microcracks, and trabecular microfractures [46, 47, 133, 134]. These three microscopic tissue damage morphologies are collectively called microdamage. Microdamage can be visualized in fluorescence microscopy by bulk staining bone specimens in one or more fluorochrome dyes including basic fuchsin, xylenol orange, calcein, and tetracyclines [131, 133, 135]. Typical microdamage parameters determined with static histomorphometry include damage volume per bone volume (DV/BV), number of microcracks per bone volume, microcrack length, and number of trabecular microfractures.

The introduction of three-dimensional imaging modalities such as micro-computed tomography (μ CT), micro-magnetic resonance imaging (μ MRI), and serial milling allowed semi-automated measurements of cortical and cancellous bone microarchitecture without the need for manual tracing by an observer or the use of a stereology model [136-140]. The three-dimensional approach to static histomorphometry provides measures of trabecular thickness, separation, bone specific surface (BS/TV), and bone volume fraction. Three-dimensional static histomorphometry does not require a trained observer to manually trace the bone in cross-section. Rather, three-dimensional static histomorphometry only requires an observer to specify a threshold to segment bone tissue from the embedding medium. A software package then automatically segments the bone image and outputs trabecular microarchitecture parameters based on voxel counts.

Three-dimensional imaging techniques also allow for novel histomorphometry measures. For example, three-dimensional static histomorphometry can determine the

structure model index of a cancellous bone specimen. Structure model index provides a description of whether a cancellous bone structure is plate-like or rod-like.

Histomorphometry can also determine the number of trabecular nodes and the degree of anisotropy. Recently, three-dimensional static histomorphometry has been expanded to include measures of microdamage. Three-dimensional imaging and analysis of microdamage has been performed using radiopaque dyes (barium sulfate and lead uranyl-acetate) with microcomputed tomography [141, 142].

3.2 Dynamic Histomorphometry

Dynamic bone histomorphometry is used to determine time-dependent measures of bone remodeling in cortical and cancellous bone [143, 144]. To perform dynamic histomorphometry, multiple injections of calcium chelating fluorochromes are administered to human or animal subjects *in vivo*. At the time of the injection, the fluorochrome binds to calcium and is incorporated within newly formed bone (Figure 3-2a). Three to ten days later, a second fluorochrome is injected (Figure 3-2b). As bone continues to form, the fluorochrome label is buried beneath the bone surface (Figure 3-2c) [145]. In human studies, tetracyclines including oxytetracycline and demeclocycline are used as labeling agents for bone formation [145, 146]. In animal studies, other compounds such as calcein and xylenol orange are often used because of their increased fluorescence intensity. However, calcein and xylenol orange are not approved for human use and are therefore limited to animal studies. Tissue processing for dynamic

histomorphometry is identical to the techniques used for traditional two-dimensional static histomorphometry [147-150].

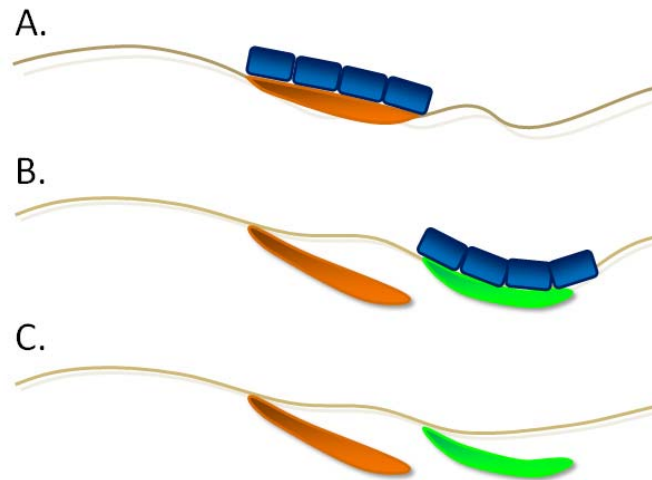


Figure 3-2. Double formation labeling is used in dynamic histomorphometry to measure bone remodeling. (A) Osteoblasts secrete osteoid on the bone surface. A fluorochrome (orange) is injected in vivo. The fluorochrome binds to any osteoid recently secreted by osteoblasts. (B) Three to ten days later, a second fluorochrome (green) is applied. (C) As bone continues to form, the fluorochrome remains beneath the bone surface.

In dynamic histomorphometry, a trained observer manually counts or traces formation label profiles using an eyepiece grid reticule. Measures of single and double labeled bone perimeter are the primary measure to determine percent mineralizing surface [148]. Measuring of mineral apposition rates require the observer to measure the distance between labels at each double labeled formation event. Most dynamic histomorphometry measurements (Table 3-2) are derived from a combination of primary static and dynamic histomorphometry measures (Table 3-1) [129]. Examples of derived dynamic histomorphometry measures include bone formation rate, formation and resorption period, BMU lifespan, etc.

Table 3-2. Derived static (structural) and dynamic (kinetic) histomorphometry measures. © (1987) Wiley. Used with permission from AM Parfitt; MK Drezner; FH Glorieux; JA Kanis; H Malluche; PJ Meunier; SM Ott; RR Recker. “Bone histomorphometry: Standardization of nomenclature, symbols, and units: Report of the asbmr histomorphometry nomenclature committee” J Bone Miner Res. Mary Ann Liebert, Inc.

Type of Index	Name of Index	Abbreviation	Formula
I. Structural	Trabecular number	Tb.N	$(BV/TV)/Tb.Th$
	Trabecular separation	Tb.Sp	$(1/Tb.N) - Tb.Th$
II. Kinetic	Mineralizing surface	MS	$(dLS+sLS/2)/BS$
	Mineral apposition rate	MAR	$Ir.L.Th/Ir.L.t$
	Adjusted apposition rate	Aj.Ar	$MAR*(MS/OS)$
	Osteoid apposition rate	OAR	Same
	Mineral formation rate	MFR	$MAR*(MS/BS)$
	Bone formation rate	BFR	Same
	Mineralization lag time	Mlt	$O.Th/Aj.Ar$
	Osteoid maturation time	Omt	$O.Th/MAR$
	Formation period	FP	$W.Th/Aj.Ar$
	Resorption period	Rs.P	$FP*(Oc.S/OS)$
	Reversal period	Rv.P	$FP*(ES-Oc.S)/OS$
	Remodeling period	Rm.P	$FP*(ES+OS)/OS$
	Quiescent period	QP	$FP*(QS/OS)$
	Total period	Tt.P	$FP*(BS/OS)$
	Activation frequency	Ac.f	$(1/Tt.P)$

3.3 Limitations of Traditional Two-Dimensional Histomorphometry

Traditional two-dimensional histomorphometry is the only method currently available for measuring bone remodeling. However, traditional histomorphometry is subject to a number of limitations. Traditional histomorphometry is time and labor intensive. Traditional histomorphometry requires tissues and features of interest to be manually traced or counted [151]. Furthermore, traditional histomorphometry is subject to considerable measurement variability between observers and laboratories [152]. Interobserver variability reduces measurement sensitivity and is of particular concern. As a result, a number of dynamic histomorphometry measurements can require impractically large sample sizes to detect differences between groups [152, 153]. For instance,

measures of the rate of appearance of new remodeling events (activation frequency, Ac.f, Table 3-2) can require 350 specimens to detect a 15% difference [152]. To mediate interobserver and interlaboratory variability, extensive training is required. Dynamic histomorphometry is also subject to errors associated with finite section thickness, reliance on a stereology model, and fluorescent bone formation labels that are out-of-plane or not observed due to excessive or insufficient time between fluorochrome injections (commonly known as label escape error) [127, 129].

Due to technical limitations, dynamic histomorphometry has been restricted to two-dimensions. Furthermore, dynamic histomorphometry measures are limited due to the irregular shape of remodeling events and cancellous bone. Because bone formation labels and resorption cavities have irregular shapes and sizes, dynamic histomorphometry cannot be used to estimate the number, surface area, or volume of individual remodeling events in cancellous bone [129, 154]. Stereology methods capable of estimating the size and number of individual objects in cancellous bone do exist. However, the techniques are impractically time and labor intensive and are not part of traditional histomorphometry [128, 129, 154, 155]. Finally, two-dimensional histomorphometry cannot determine whether features within cancellous bone are spatially associated with one another. For example, previous work using two-dimensional histomorphometry suggests that microdamage may be associated with regions of high tissue stress [156, 157]. However, the limitations of two-dimensional histomorphometry render the results of this study inconclusive.

The inability of dynamic histomorphometry to measure individual remodeling events presents an important limitation. Dynamic histomorphometry cannot distinguish

between multiple small remodeling events and one very large remodeling event. The two cases are distinctly different biologically and, likely, mechanically [7, 158]. The ability to measure the number, breadth, and depth of resorption cavities and formation events would provide valuable information that could improve our understanding of the influence of bone cell biology on bone remodeling and biomechanics [158]. A three-dimensional approach to dynamic bone histomorphometry would allow for measures of individual remodeling events. Furthermore, three-dimensional dynamic histomorphometry could determine the spatial relationship between individual remodeling events, local tissue stresses, and microdamage. One imaging approach capable of visualizing bone tissue and fluorescent labels in three-dimensions is serial milling.

4 Serial Milling with Epifluorescence Imaging

This chapter was published in the Journal of Microscopy, Volume 236 number 1. The article is titled “Three-dimensional surface texture visualization of bone tissue through epifluorescence-based serial block face imaging” by Slyfield, CR, Niemeyer, KE, Tkachenko, EV, Tomlinson, RE, Steyer, GG, Patthanacharoenphon, CG, Kazakia, GJ, Wilson, DL, and Hernandez CJ, p. 52-59, Copyright (2009) and is reprinted here with permission of John Wiley and Sons.

4.1 Abstract

Serial block face imaging is a microscopy technique in which the top of a specimen is cut or ground away and a mosaic of images is collected of the newly revealed cross-section. Images collected from each slice are then digitally stacked to achieve three-dimensional images. The development of fully-automated image acquisition devices has made serial block face imaging more attractive by greatly reducing labor requirements. The technique is particularly attractive for studies of biological activity within cancellous bone as it has the capability of achieving direct, automated measures of biological and morphological traits and their associations with one another. When used with fluorescence microscopy, serial block face imaging has the potential to achieve 3D images of tissue as well as fluorescent markers of biological activity. Epifluorescence-based serial block face imaging presents a number of unique challenges for visualizing bone specimens due to noise generated by subsurface signal and local variations in tissue

autofluorescence. Here we present techniques for processing serial block face images of trabecular bone using a combination of non-uniform illumination correction, precise tiling of the mosaic in each cross-section, cross-section alignment for vertical stacking, removal of subsurface signal, and segmentation. The resulting techniques allow examination of bone surface texture that will enable 3D quantitative measures of biological processes in cancellous bone biopsies.

4.2 Introduction

Bone remodeling is the primary processes for modifying bone mass in adults. Bone remodeling is therefore a key process in the development and treatment of osteoporosis and other metabolic bone diseases. Bone remodeling occurs at discrete locations on the bone surface and consists of a group of osteoclasts that remove a patch of bone, followed in time by a group of osteoblasts that form bone at the same location. Characteristics of bone remodeling such as the rate of appearance of new remodeling events, the percent eroded surface (where osteoclasts have resorbed bone and formation has not yet initiated), and the rate at which new bone is formed are often measured histologically in bone biopsies to diagnose disease and to compare treatments. Histological examination of bone biopsies remains the only way to measure localized aspects of bone remodeling [159]. Currently, such histological analysis is performed on two-dimensional sections using light and epifluorescence microscopy as part of a technique known as “dynamic bone histomorphometry” [129, 148, 160]. Dynamic bone histomorphometry is a specialized technique that uses model-based stereological

principles to obtain quantitative data about the bone remodeling process in a specimen. Dynamic histomorphometry is commonly performed in clinical and pre-clinical studies of osteoporosis therapies and occasionally for diagnosis of metabolic bone disease. In a dynamic bone histomorphometry study, subjects are given two doses of a fluorescent marker separated by some time (typically one week in human studies) [148, 149]. The fluorescent marker (tetracycline in humans) binds to regions of newly forming bone. In the histology slide, the percent of bone surface covered with the fluorescent label, and the distance between the two labels are used to calculate measures of bone formation and the rate of bone turnover (commonly expressed per unit bone surface). Additionally, the eroded surface is measured as an assay of osteoclast activity. These dynamic bone histomorphometry measures are commonly performed through stereological point counting as well as by manually tracing regions of interest using commercially available microscope-computer data acquisition systems. Surface areas in bone, some of the most important types of measures, are commonly estimated from perimeter measurements using a geometric model [129].

Although dynamic bone histomorphometry is the only technique capable of achieving local measures of bone remodeling, the approach is not without limitations. After the technique is applied to two-dimensional histological sections, a geometric model is commonly used to interpret two-dimensional measures of perimeter in terms of surface area [129]. Another challenge in dynamic bone histomorphometry is that many of the measurements require manual segmentation and tracing of traits of interest, making the analysis technique time intensive and subject to considerable inter-observer variation, factors that limit precision [129, 151, 161, 162]. Lastly, the spatial relationship between

different aspects of the remodeling process (distances between bone formation and bone resorption for example) cannot be evaluated in two-dimensional sections and is an unproven assumption used in dynamic bone histomorphometry. Unbiased model-independent methods have been occasionally applied to remove the need for a geometric model when evaluating percent labeled and percent eroded surfaces [163].

Instead of dynamic histomorphometry, we used a direct 3D approach. Serial block face imaging is an image acquisition technique that has the potential to lead to three-dimensional dynamic bone histomorphometry. The technique has been used to obtain three-dimensional images of cancellous bone structure for some time [164-166]. Serial block face imaging involves repeatedly trimming off the top of a specimen and collecting images of the specimen block's newly revealed cross-section. By collecting images of the specimen block instead of serial sections, labor requirements are reduced because there is no need to mount and align individual sections. Additionally, because a mosaic of images can be collected using serial block face imaging, the size of the specimen that can be imaged is independent of resolution, unlike other common three-dimensional imaging techniques, such as micro-computed tomography, in which the field-of-view decreases as image resolution increases and the capability of imaging multiple overlapping regions to form mosaics may not be present. While confocal microscopy is also capable of obtaining three-dimensional images like those obtained through serial milling, it is severely limited in the depth of material that may be visualized (~100 μm maximum). A limitation of serial block face imaging is that it is destructive (no histological slides are available for later analysis) and time intensive [166].

Recently, Kazakia and colleagues demonstrated a technique for serial block face imaging implemented through fully-automated serial milling [166]. Images were collected using fluorescent microscopy and were shown to be suitable for forming three-dimensional images of bone and fluorescent markers, indicating that the image acquisition approach has the potential to achieve three-dimensional images for dynamic bone histomorphometry. However, bone formation indices (i.e. fluorescent markers) are not the only important measures in bone histomorphometry, measures of the bone surface are also important (measures of fluorescent markers are expressed relative to bone surface). Additionally, measures of the percent of the bone surface that has been eroded by osteoclasts are also used to interpret histomorphometry data. Eroded surfaces are regions on the bone that have undergone or are currently undergoing bone resorption and are detected as irregularities on the bone surface (the so-called “scalloped surface” or “crenated surface”) [129]. Eroded surfaces are not readily visible in images obtained by Kazakia and colleagues, suggesting that increased image resolution and/or improved image processing techniques are required to detect such surface traits. Potential limitations of previously used image processing techniques are that they do not account for noise generated by material below the surface of the specimen or variability in tissue intensity, which can make segmentation challenging.

The long-term goal of this work is to achieve direct three-dimensional measures of bone remodeling activity in bone biopsies. Toward that goal, the current work demonstrates a method of processing images obtained through serial block face imaging to capture representations of bone surface texture and architecture that can be used to identify eroded surfaces.

4.3 Methods

4.3.1 Imaging System

Serial block face imaging using epifluorescence is implemented using a serial milling device as described by Kazakia and colleagues [166]. We refer the reader to the original publication for a thorough description of the image acquisition device and comparison to other available approaches. We only briefly describe the technique here. The device consists of a computer numerically controlled (CNC) milling device with an attached microscope, high-resolution digital camera, fluorescent light source and automated filter changer. A specimen embedded in methyl-methacrylate is secured to the CNC workbench and the top five micrometers of the sample are removed by the endmill (Figure 4-1). Automated air pressure release and vacuum ensure that the surface is clear of cutting particles. Cutting artifacts on the specimen surface (grooves aligned with endmill rotation) are present but have a negligible effect on image clarity. After cutting, the specimen is automatically positioned beneath the microscope and images are obtained using one or more fluorescence filter sets. The process is repeated until the desired volume of the specimen has been acquired. In our implementation, a 14 bit gray-scale digital camera with 2048 X 2048 pixel field-of-view is used to collect images (pco2000, Cooke Corp., Romulus, MI, USA). Resolution of the images is controlled using different microscope objectives (we have achieved success with 4X and 10X objectives, corresponding to pixel sizes ranging from 1.8 μm to 0.7 μm). Images described below are

collected using a 10X objective ($0.7\mu\text{m}$ / pixel). Out-of-plane resolution is determined by the amount of material trimmed away in each slice ($5\mu\text{m}$ in the current implementation). A complete raw image of rat vertebral cancellous bone requires a 4 X 3 array of images within each cross-section for a total cross-sectional area of 5.1 X 4.4 mm. A depth of 4mm achieved using 800 image slices ($5\mu\text{m}$ / slice) requires five days of fully-automated image acquisition. The raw images occupy 80 GB of memory for each fluorescent channel (16-bit data).



Figure 4-1. Serial fluorescence microscopy is accomplished using a CNC milling device. An attached grayscale camera, automated filterset changer, and microscope objective lens are used to serially and automatically acquire images of each cross-section.

4.3.2 Image Processing Overview

A series of automated image processing steps is used to generate three-dimensional images of the bone surface. First, non-uniform illumination correction is applied to the images to equalize the contrast and intensity of the signals over each image

field-of-view (FOV). Subsequently, a mosaic is created by aligning overlapping image regions using normalized cross-correlation. Alignment (registration) of each cross-section in the stack is performed using fiduciary markers (vertical objects placed close to the specimen to allow more precise alignment for stacking). Cross-sectional alignment then allows for precise subtraction of sub-surface fluorescence. Each image cross-section is then cropped to present only cancellous bone from the rat vertebra, reducing image size for the 800 slices from 80 GB/fluorescent channel to 35 GB/fluorescent channel (stored as 16-bit data). Finally, segmentation is performed using an advanced technique known as iterative thresholding [167]. The segmented signals occupy a total of 2.2 GB of data and can then be visualized as isosurfaces. Image processing is performed using custom software written for Matlab (Mathworks, Natick, Massachusetts, USA). Three-dimensional visualization is performed using Amira (Visage Imaging, Inc., Carlsbad, California, USA).

4.3.3 Non-uniform Illumination Correction

Non-uniform illumination in epifluorescence imaging results in reduced signal intensity towards the edges and corners of each field-of-view. Non-uniform illumination is problematic because reduced pixel intensities at image edges result in decreased signal-to-noise ratio. Poor signal-to-noise ratio can result in segmentation errors and prevent proper mosaic tiling. To compensate for non-uniform illumination, there are two approaches. The first is to use a smaller field-of-view centered on the light source to achieve a flat illumination profile. One drawback to this option in serial block face

imaging is that a reduced field-of-view requires a larger mosaic (more images per cross-section) to capture the same image area, substantially increasing image acquisition time. Instead, we compensate for non-uniform illumination post-hoc.

Background intensity profiles for our system are well characterized by polynomial surfaces [168]. The non-uniform illumination correction algorithm we employ uses a second order polynomial intensity profile fit in the x-y plane of the form:

$$I_B = a_0 + a_1x + a_2y + a_3x^2 + a_4y^2 + a_5xy \quad (1)$$

To determine the coefficients a_i , a system of linear equations is solved using the intensities (I_B) at locations within an image. The locations are taken from the lowest intensity values in each of 16 equally sized rectangular subdivisions of the image. Although only six points are necessary to solve (1), all 16 are used to reduce the contribution of noise. Using these parameters, an illumination profile is generated for each image in the mosaic. The original image is then divided by the illumination profile (Figure 4-2) [169, 170].

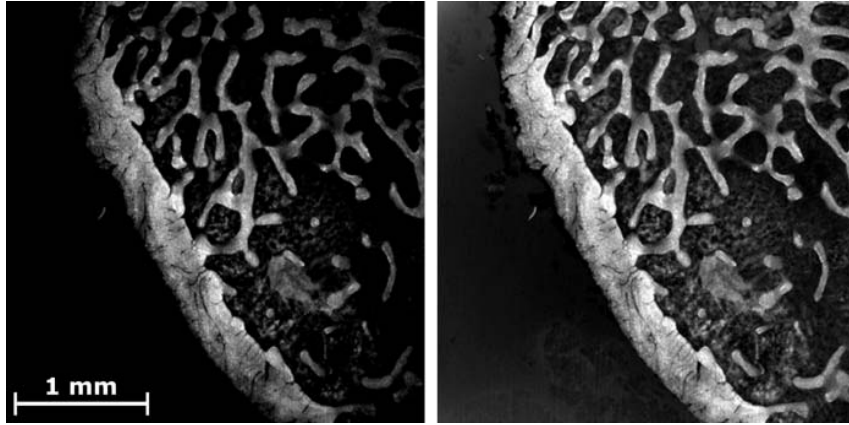


Figure 4-2. An image of a bone specimen (bright) is shown. (Left) The original image shows reduced signal intensity at the periphery of the FOV. (Right) The same image after correction for non-uniform illumination.

4.3.4 Mosaic Tiling

An important advantage of using serial block face imaging is that a mosaic of images can be collected. This allows image resolution to be selected independent of the physical size of the specimen. When collecting a mosaic of images, the repeatability of stage positioning can affect the relative positions of each field-of-view. This effect is negligible when the positioning repeatability of the stage is less than one image pixel. However, at higher magnification, this inaccuracy has the potential to influence quantitative measurements of surface texture, surface area and spatial correlations among small features.

Because our pixel size is much smaller than the positioning repeatability in our system (5 μm), images in the mosaic must be aligned using automated tiling [169]. To reduce computational expense, only regions corresponding to a prescribed overlap \pm camera positioning repeatability are used to achieve the tiled mosaic. Vertical alignment of each cross-section is similarly subject to stage positioning inaccuracy and is addressed

using a set of fiduciary markers (in this case perpendicular holes drilled into the embedding media near the specimen) [15].

4.3.5 Sub-surface Signal Removal

Serial block face imaging with epifluorescence is subject to noise created by signal originating below the primary optical plane. The sub-surface signal manifests itself as blurred features in an image (Figure 4-3). The use of an opaque embedding medium can mediate this effect, but subsurface fluorescence cannot be attenuated completely. To remove a majority of sub-surface fluorescence, we first express the intensity of light detected by the imaging system in two consecutive slices as a sum of the intensity within the optical plane as well as that below the optical plane as follows:

$$I_i = (I_o T_{at}) F_i T_{ta} + (I_o T_{at} e^{-\mu s}) F_{i+1} (e^{-\mu s} T_{ta}) + (I_o T_{at} e^{-2\mu s}) F_{i+2} (e^{-2\mu s} T_{ta}) + \dots \quad (4)$$

$$I_{i+1} = (I_o T_{at}) F_{i+1} T_{ta} + (I_o T_{at} e^{-\mu s}) F_{i+2} (e^{-\mu s} T_{ta}) + (I_o T_{at} e^{-2\mu s}) F_{i+3} (e^{-2\mu s} T_{ta}) + \dots \quad (5)$$

where I_o is the excitation light intensity. I_i and I_{i+1} represent the intensity of a pixel measured in two sequential cross-sections. The F_i terms represent the actual fluorescence intensities of tissue at each slice number i . The terms μ and T_{at}/T_{ta} , which vary from 0 to 1, represent coefficients of signal attenuation due to embedding medium and the transition from air to tissue/tissue to air, respectively and s is the thickness of each slice (5 μm in this application). If we assume that all sub-surface signal seen in an image is

contained within image $i+1$, subtraction of the two observed intensities will obtain the true in-plane signal (Figure 4-3). The true signal is then calculated as follows:

$$I_i - e^{-2\mu s} I_{i+1} \otimes h(x,y) = I_o T_{at} F_i T_{ta} = \text{True Signal}, \quad (6)$$

where h represents a convolution kernel simulating the dispersion of light due to the scattering effect of the tissue and embedding medium. A parametric analysis found that Gaussian kernel of size 5×5 provides suitable results, as it rather accurately models the blurring due to the dispersion effect of the embedding medium. The attenuation constant of the scattered signal (μ) is calculated from the signal intensity of pixels that are identified as partial volumes (i.e. where the specimen is present in one slice but not the slice below). The value of μ is specific to each specimen due to variations in the opacity of the embedding medium from specimen to specimen. Typical values of μ were 0.04 – 0.08 in the specimens examined in the current study (note that T_{at} and T_{ta} do not need be explicitly determined).

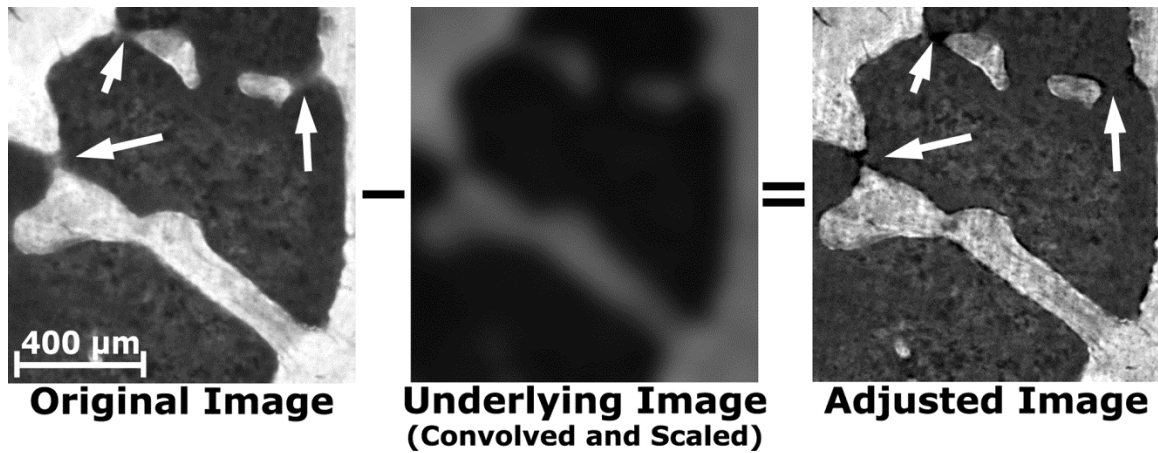


Figure 4-3. The original image of trabecular bone shows considerable sub-surface signal observed as blurring (Arrows). The underlying image is convolved, scaled, and then subtracted from the original image to adjust for sub-surface signal. The adjusted image does not have the blurred regions and the edges of the bone are more defined.

Deconvolution can be used as a secondary method to remove sub-surface signal and will also remove blurring due to lens aberrations. Application of 3D deconvolution in our implementation caused only marginal improvements in out-of-plane signal removal and did not justify the computational expense.

4.3.6 Segmentation

Segmentation and isosurface renderings are required for automated and semi-automated measures of bone remodeling in 3D images. Global thresholding can be insufficient for segmenting epifluorescence images of bone due to variations in signal intensity and noise associated with autofluorescence of neighboring tissues (marrow cells and extracellular matrix). A liberal global threshold is capable of retaining most, if not all, of the bone in the segmentation, but the resulting segmentation will often contain neighboring tissues, potentially confounding measures of bone surface texture. Similar

challenges are present when segmenting images of bone collected through micro computed tomography where global thresholds are typically set manually, introducing inter-observer variation [171]. Images with strong edges (high contrast) and relatively uniform signal intensity (specimen/void interfacial gradients) can successfully be segmented using local adaptive algorithms, such as that proposed by Burghardt *et al.* [172]. However, images collected through epifluorescence (particularly images of bone) require a different segmentation approach.

One solution is to employ an "iterative" thresholding algorithm as described by Wu *et al.* [167]. Although we refer the reader to the original article for a complete description of the iterative thresholding approach, we will describe it briefly here. In iterative thresholding, a field of pixel specific threshold values is iteratively adjusted based on the local neighborhood (Figure 4-4). A matrix containing pixel specific threshold values (T_{xy}) is created and applied to the original image. This threshold matrix is initially a uniform value determined using the histogram triangle method [173]. A second matrix (ΔT_{xy}) is then created to describe a change in pixel specific threshold values. ΔT_{xy} is created such that the threshold T_{xy} will be reduced for pixels that exceed the threshold and will be increased for pixels that are below the threshold. ΔT_{xy} is then convolved by a 3 x 3 Gaussian kernel so that changes in the threshold are mediated by the local neighborhood. ΔT_{xy} is then added to T_{xy} to create new pixel-specific threshold values. These steps are repeated a specified number of iterations. The total number of iterations is selected based on comparisons of the segmented image to the grayscale version, similar to selections of global thresholds used in processing micro-computed tomography images [174]. In our implementation 25-30 iterations are commonly selected

(Figure 4-5). Following segmentation, a morphological processing step is applied in 3D to remove any remaining groups of free-floating voxels that represent noise in images of bone (from components within the marrow space). Total image processing time for a rat lumbar vertebra (5.1mm X 4.4mm X 4mm) is typically 5 days on a 2.66GHz dual-core Intel processor with 1333 MHz front side bus and 18GB of RAM.

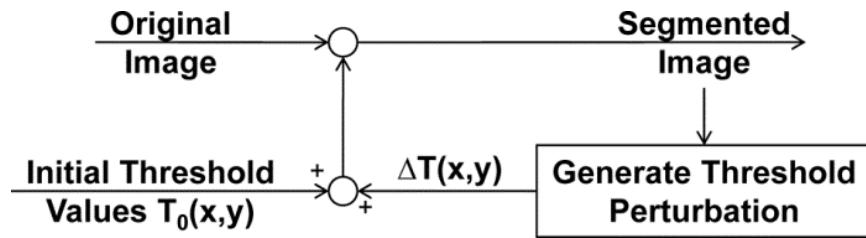


Figure 4-4. Images are segmented using an iterative thresholding algorithm adapted from Wu, *et al.* [167]. An original image is segmented using a matrix of pixel-specific threshold values. Subsequently, the threshold values are adjusted and the original image is segmented again. This process is repeated for a number of iterations selected by user comparison to the gray-scale image.

To determine improvements gained by using these steps, images of cancellous bone from the fourth lumbar vertebra from five rats are collected and processed using the same approach used in previous publications (no mosaic tiling or out-of-plane signal subtraction and a global threshold) as well as the current technique. The two techniques are compared by measuring bone volume fraction (the ratio of bone volume to total image volume, BV/TV), specific surface (ratio of bone surface to total image volume, BS/TV), and the mean local surface curvature across the entire bone surface. Mean local surface curvature is used as an indicator of how well the processing techniques represent roughness of the final bone surface. Local curvature measures are obtained using a built-

in function in Amira (GetCurvature) that fits a quadric surface to a 26-connected local neighborhood around each polygon. The maximum eigenvalue (in magnitude) for each quadric surface is determined and the average over the whole specimen is used as a measure of local curvature, with larger values indicating more irregular surface texture. Because scanning electron microscopy of cancellous bone suggests that bone surfaces not part of resorption cavities are smooth [175], images with lower mean local curvature values can be considered more accurate depictions of the bone surface (eroded surfaces typically represent less than 10% of the total bone surface and will therefore not greatly influence measures of local surface curvature) [176]. The raw images, collected at a resolution of $0.7\text{ }\mu\text{m}$ / pixel in-plane, are uniformly coarsened to $3.6\text{ }\mu\text{m}$ / pixel in-plane to determine the influence of resolution on the final image. Statistical comparisons of images processed using different techniques were performed using paired t-tests.

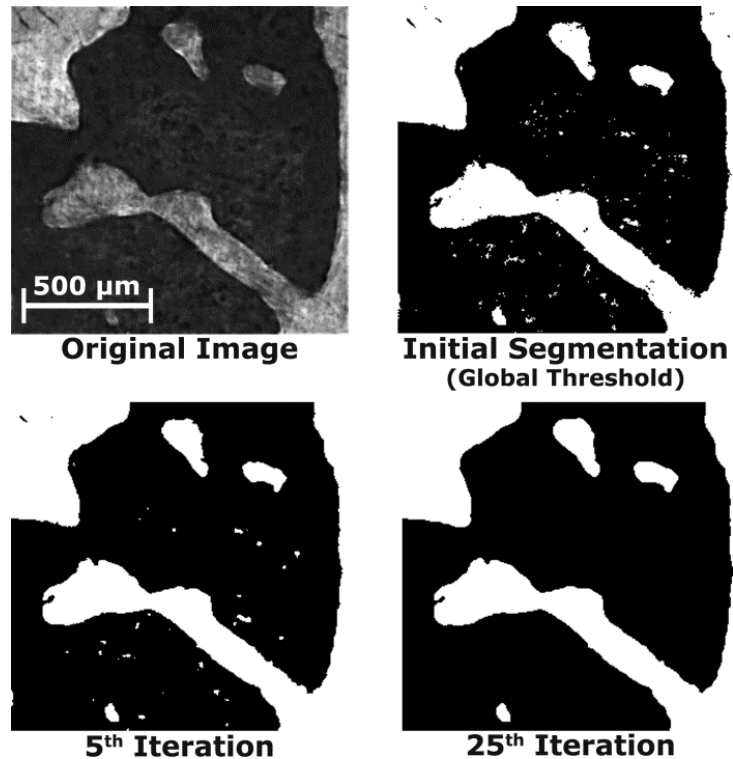


Figure 4-5. An original image shows bone (bright) and surrounding marrow (dark). More bone and less marrow is included in the segmentation as the algorithm iterates (bottom left). After the final (25th) iteration, some free floating pixels representing marrow components (noise) are removed (bottom right). Note the improvement in the segmentation in comparison to global thresholding (top right).

4.4 Results

High-resolution images processed with the techniques described in the current study showed much smoother bone surfaces than those processed with previous techniques (Figs. 4-6, 4-7, 4-8). Surface irregularities characteristic of osteoclast activity are readily visible in the grayscale and segmented images at 0.7 μm resolution (Figs. 4-7, 4-8) and are not visible in lower resolution images or images processed using previous techniques. Iterative thresholding was particularly effective because it properly segmented noise and was capable of producing a segmentation in which the irregular

surface texture resulting from remodeling activity was distinguishable from the rest of the bone.

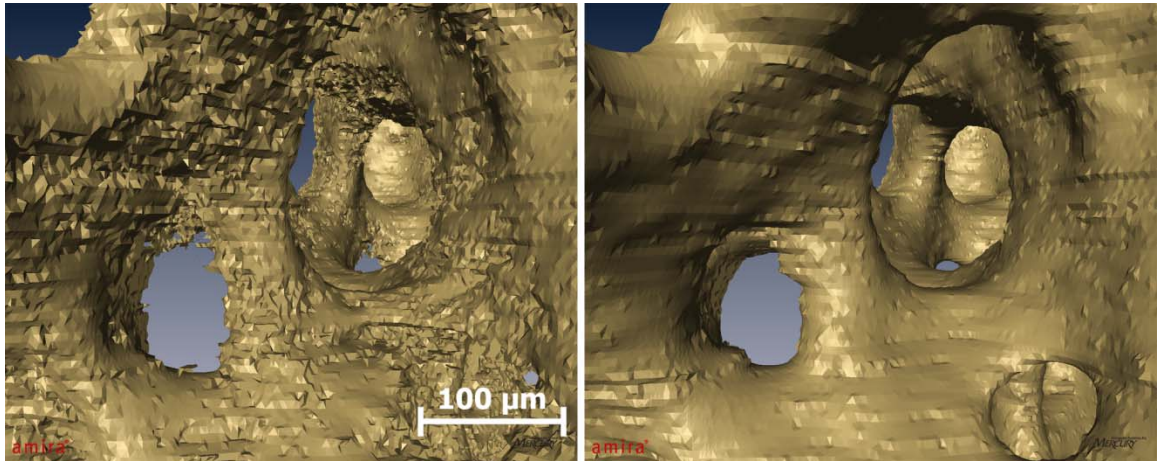


Figure 4-6. Images of cancellous bone acquired through serial block face imaging processed with previously described techniques (left) and with the current techniques (right). Both images have been visualized using identical isosurface smoothing parameters. Images processed using the current technique show a much smoother bone surface that is similar to images obtained using scanning electron microscopy [175].

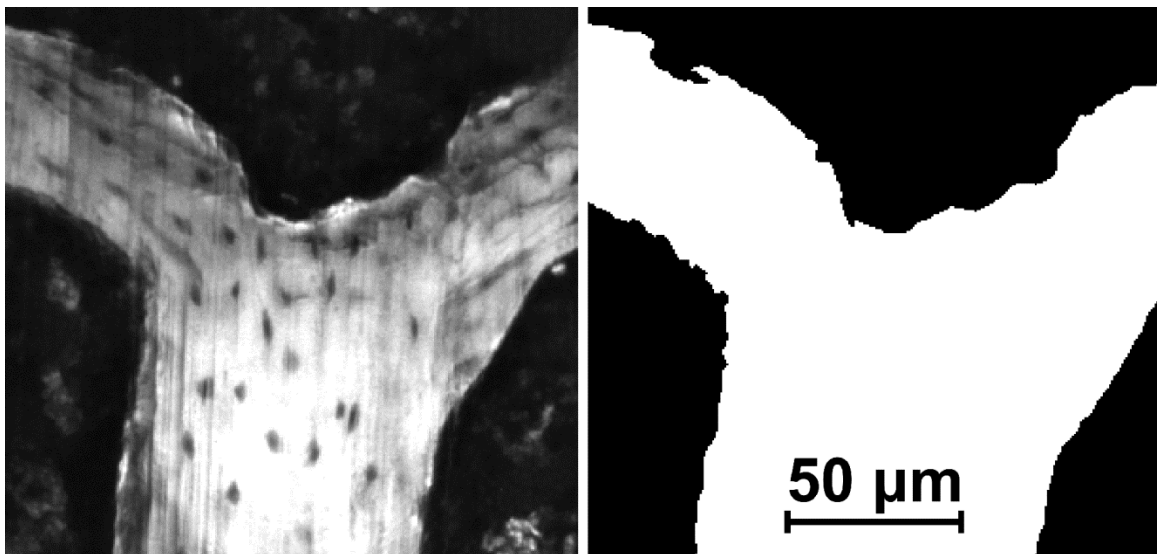


Figure 4-7. (Left) A gray scale image of the cancellous bone surface collected using serial block face imaging is shown. An eroded surface is indicated by bone surface irregularities (arrows). Non-eroded surfaces are smooth (lower right corner of the image). (Right) A fully processed version of the same image in which the bone surface irregularities characteristic of eroded surface can also be seen.

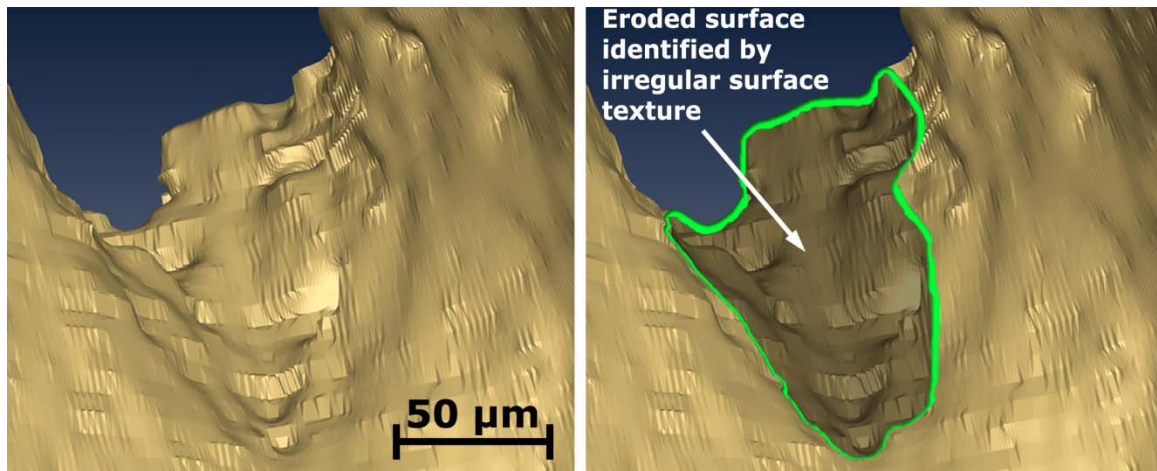


Figure 4-8. An isosurface rendering of a location on the bone surface is shown illustrating surface texture variations characteristic of eroded surfaces.

In the highest resolution images, the inclusion of automated tiling, out-of-plane signal subtraction, and iterative thresholding resulted in a large reduction in mean surface curvature. Images processed using the current techniques with an in-plane resolution of $0.7\ \mu\text{m}$ showed a mean surface curvature 40.6% less than when processed using previous techniques ($p < 0.05$, Table 4-1). No significant differences in bone volume fraction or bone specific surface (the ratio of bone surface to total specimen volume) were observed among images at $0.7\ \mu\text{m}/\text{pixel}$ resolution. In images with an in-plane resolution of $3.6\ \mu\text{m}$ (similar to the resolution used in previous serial milling implementations) no significant differences in mean surface curvature, bone volume fraction or specific surface were observed between processing methods.

Table 4-1. Bone volume fraction, bone surface density, and the mean surface curvature of cancellous bone isosurfaces (mean \pm SD) were measured in 5 rat lumbar vertebrae using the previous image processing techniques and the revised techniques presented in the current study. Non uniform illumination (NUI) correction was applied in all cases.

	0.7 μm Previous	0.7 μm Revised	3.6 μm Previous	3.6 μm Revised
Bone Volume Fraction (BV/TV)	35.6 \pm 3.5%	35.2 \pm 3.3%	37.0 \pm 5.5%	35.1 \pm 3.3%
Bone Specific Surface (BS/TV, 1/mm)	12.8 \pm 1.3	13.8 \pm 2.6	11.7 \pm 2.0	11.6 \pm 1.7
Mean Surface Curvature (1/ μm)	2.6 \pm 0.6	1.8 \pm 0.5*	1.1 \pm 0.6	1.3 \pm 0.5

* Different from image processed using previous methods ($p < 0.05$, paired t-test)

4.5 DISCUSSION

At high resolution (0.7 μm / pixel in-plane), the image processing techniques described here provide a substantial improvement in bone surface representation. At lower resolutions, however, there is little improvement in surface representation when using the more refined processing we have presented. Furthermore, measures of bone volume fraction do not appear to be sensitive to image resolution and are similar when using both image processing techniques.

The strength of the current image processing approach is that it addresses the major challenges in processing images obtained through serial block face imaging using fluorescent microscopy: mosaic tiling and vertical alignment, out-of-plane signal, and variations in tissue intensity. Additionally, when using the technique on high-resolution (0.7 μm / pixel) images, eroded surfaces are visible and appear similar to those observed

using scanning electron microscopy. This approach therefore has the potential to allow measurements of the number and size of individual eroded surfaces in a three-dimensional image of cancellous bone, measurements that cannot be achieved with traditional two-dimensional histomorphometry techniques. A limitation of the current approach is that the image processing methods are specific to serial block face imaging.

Previous implementations of serial block face imaging through epifluorescence used lower image resolution ($3.3\text{ }\mu\text{m}$ / pixel in-plane) and more rudimentary image processing techniques (no adjustment for out-of-plane signal and use of a global threshold) [166]. The current study suggests that such an approach will achieve adequate representations of bone microarchitecture (as are commonly performed in three-dimensional images obtained through micro-computed tomography), but will not provide representations of bone surface texture needed to detect eroded surfaces. At higher image resolutions ($0.7\text{ }\mu\text{m}$ / pixel in-plane) surface texture irregularities characteristic of eroded surfaces can be detected, but only when using more complicated image processing steps, such as those presented in the current study. Serial block face imaging has a number of advantages over other three-dimensional imaging modalities that may be applied to bone (confocal microscopy, micro-computed tomography, serial sectioning) because it can capture both an almost unlimited specimen size and fluorescent markers (see Kazakia *et al.* 2007 for a thorough discussion of the advantages and disadvantages of this image acquisition approach). Although analysis of fluorescent markers of bone formation was not performed in the current study, the same image processing techniques could be applied to images of bone with fluorescent markers of bone formation (Figure 4-9). With the inclusion of bone formation markers, unique measurements such as the spatial

relationship between eroded surfaces and local bone formation could be achieved. The advancements in processing presented here will enable further development of automated and semi-automated measurement techniques to achieve three-dimensional dynamic bone histomorphometry. Future studies will involve validation of these three-dimensional measurements, including comparison to the current two-dimensional histology standard.

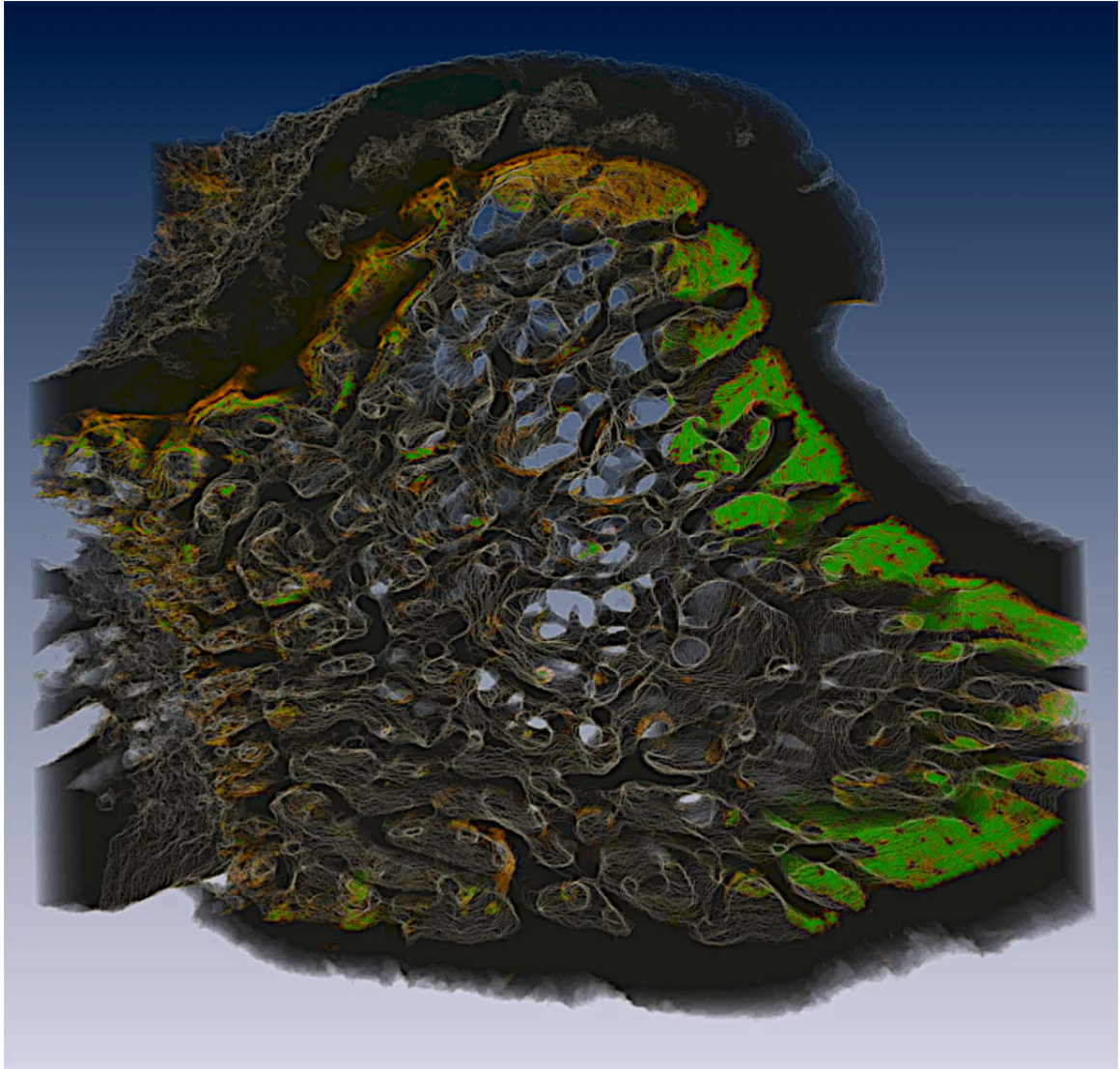


Figure 4-9. An image of a rat tibial metaphysis, collected using serial milling and processed using the methods described in the current study, is shown. Two different fluorescent markers were applied to the animal prior to euthanasia, xylene orange (shown in orange) followed one week later by calcein (shown in green). Three-dimensional images of the bone and the fluorescent markers of bone formation were collected using serial block face imaging with fluorescent markers processed using global thresholding. Future work will refine and validate the image processing techniques used to examine the fluorescent markers to allow direct, three-dimensional measures of eroded surfaces as well as the fluorescent markers of bone formation.

4.6 ACKNOWLEDGEMENTS

This work was supported by NIH/NIAMS R21 AR054448 and T32 AR007505-22 (C.R.S.). Dr. Wilson has an interest in BioInVision, Inc., which may commercialize the serial milling technology.

5 Three-Dimensional Dynamic Bone Histomorphometry

This chapter is currently in press in the Journal of Bone and Mineral Research. The article is titled “Three-dimensional dynamic bone histomorphometry” by Slyfield, CR, Tkachenko, EV, Wilson, DL, and Hernandez CJ. Copyright (2011) and is reprinted here with permission of John Wiley and Sons.

5.1 Abstract

Dynamic bone histomorphometry is the standard method for measuring bone remodeling at the level of individual events. While dynamic bone histomorphometry is an invaluable tool for understanding osteoporosis and other metabolic bone diseases, the technique's two-dimensional nature requires the use of stereology and prevents measures of individual remodeling event number and size. Here, we use a novel three-dimensional fluorescence imaging technique to achieve measures of individual resorption cavities and formation events. We perform this three-dimensional histomorphometry approach using a common model of postmenopausal osteoporosis, the ovariectomized rat. The three-dimensional images demonstrate the spatial relationship between resorption cavities and formation events consistent with the hemi-osteonal model of cancellous bone remodeling. Established ovariectomy was associated with significant increases in the number of resorption cavities per unit bone surface ($2.38 \pm 0.24 \text{ mm}^{-2}$ SHAM v. $3.86 \pm 0.35 \text{ mm}^{-2}$ OVX, mean \pm SD, $p < 0.05$) and total volume occupied by cavities per unit bone volume ($0.38 \pm 0.06\%$ SHAM v. $1.12 \pm 0.18\%$ OVX, $p < 0.001$), but no difference in surface area

per resorption cavity, maximum cavity depth, or cavity volume. Additionally, we find that established ovariectomy is associated with increased size of bone formation events due to merging of formation events ($23,700 \pm 6,890 \mu\text{m}^2$ SHAM v. $33,300 \pm 7,950 \mu\text{m}^2$ OVX). No differences in mineral apposition rate (determined in 3D) were associated with established ovariectomy. That established estrogen depletion is associated with increased number of remodeling events with only subtle changes in remodeling event size suggests that circulating estrogens may have their primary effect on the origination of new basic multicellular units with relatively little effect on the progression and termination of active remodeling events.

5.2 Introduction

Bone remodeling is the primary process of modifying bone volume and microstructure in adults and is therefore a key process in the development and treatment of osteoporosis. Bone remodeling is a focal process that occurs at discrete locations on bone surfaces. An individual remodeling event is known as a basic multicellular unit (BMU). A basic multicellular unit consists of a group of osteoclasts that resorb bone followed in time by a group of osteoblasts that form bone [85]. In cancellous bone, the basic multicellular unit is believed to form a trench-like remodeling cavity on the bone surface referred to as a hemi-osteon [177]. The birthrate, number, and size of remodeling events determine the overall rate of bone turnover [86]. Because bone resorption precedes formation, each remodeling event is associated with the creation of a resorption cavity (Howship's lacuna). Resorption cavities on the surfaces of trabeculae are believed to

generate stress concentrations that have a detrimental effect on the biomechanical performance of the entire cancellous bone structure [178]. Simple biomechanical analyses suggest that differences in the number and size (breadth and depth) of individual resorption cavities influence bone biomechanics [179, 180].

Dynamic bone histomorphometry is the standard method for evaluating alterations in bone remodeling at the level of the basic multicellular unit [143, 144]. Dynamic bone histomorphometry is unique in its ability to measure parameters such as the mineral apposition rate, bone formation rate and activation frequency [129]. In addition to measures of bone formation, bone histomorphometry studies typically include measures of bone volume fraction and percent eroded surface (ES/BS). Dynamic bone histomorphometry has provided key information regarding the development and treatment of osteoporosis and other metabolic bone diseases. However, measures of the size and number of remodeling events are not performed in traditional dynamic bone histomorphometry as they require labor intensive serial sections or specialized stereology techniques (the 'disector method') [128, 129, 155].

In our laboratory, we have advanced an imaging modality known as serial block face imaging [164, 165]. Serial block face imaging can achieve three-dimensional images of bone and fluorescent markers of bone formation at a sub-micrometer scale [166, 181]. We have previously demonstrated and validated approaches for visualizing bone surface texture and obtained measures of the number and surface size of resorption cavities (identified as eroded surfaces) in three-dimensional images [182]. Here we demonstrate an approach for three-dimensional measures of bone formation as well as resorption cavities and apply the approach to ovariectomized rats, the most commonly used animal

model of post-menopausal osteoporosis.

Ovariectomy-induced estrogen depletion results in an increase in bone turnover in the rat, indicated by increases in mineralizing surface, osteoid surface, and eroded surface in cancellous bone present months after surgery [183, 184]. It has been proposed that the increase in bone remodeling in established estrogen depletion is due to an increase in the birthrate of new remodeling events and extended progression of individual basic multicellular units across the bone surface, increasing both the number and surface area of individual remodeling events [177, 185, 186]. Traditional histomorphometry has provided detailed accounts of ovariectomy-induced estrogen depletion [187]. However, because remodeling events have not been observed and quantified in cancellous bone in three-dimensions, it is not known if established estrogen depletion consists of an alteration in the number or size of remodeling cavities; a subtle difference in bone remodeling that can alter local stress concentrations [179, 180, 188, 189]. Additionally, it is not known how three-dimensional measures of bone formation (number and size of forming surfaces, mineral apposition rate, etc.) vary among remodeling sites or in established ovariectomy.

The overall goal of this line of investigation is to improve assessment of bone remodeling at the level of the individual remodeling event. We hypothesize that established estrogen depletion results in an increase in the number and size of resorption cavities and sites of bone formation. To test this hypothesis, we: 1) establish image processing and analysis methods to create three-dimensional images of bone formation markers and measures of resorption cavity depth and volume; and 2) determine the effect of established ovariectomy-induced estrogen depletion on the number and size of

resorption cavities (identified as eroded surfaces), formation events, and mineral apposition rate at each formation event.

5.3 Methods

5.3.1 Image Acquisition

Ten, six-month-old female Sprague-Dawley rats were subjected to bilateral ovariectomy (OVX) or sham surgery (n=5 per group). Four months after surgery, the animals were euthanized and the fourth lumbar vertebrae were collected (these animals were part of a larger study examining drug treatment following established ovariectomy). This age group of animals was selected to characterize the condition of established estrogen depletion, after a new, post-ovariectomy steady state in bone remodeling is achieved. Animals were subjected to two doses of bone formation markers at 10 and 3 days prior to euthanasia (90 mg/kg xylenol orange followed by 10 mg/kg calcein). All animal use was approved by the Case Western Reserve University IACUC. Following euthanasia, the fourth lumbar vertebrae were dissected to remove soft tissue. The posterior elements and endplates of each vertebra were removed with a low speed diamond saw (Buehler Corp., Lake Bluff, IL, USA) and bone marrow was washed out using a low-pressure water jet. Vertebrae were fixed in 10% buffered formalin and embedded undecalcified in opaque methyl-methacrylate.

Three-dimensional images of each specimen were collected using serial block face imaging implemented through fully-automated serial milling. We present a brief

description of the approach here, but refer the reader to prior publications for additional details [166, 181]. Serial milling involves repeatedly trimming away (destructively) the top of a specimen and collecting a mosaic of images of the newly revealed surface. The images are then stitched together and stacked to achieve three-dimensional images of each specimen at a resolution of $0.7 \times 0.7 \times 5.0$ micrometers per voxel. Three images of each specimen cross-section were collected using different fluorescent filter sets: one channel to visualize bone tissue autofluorescence (UV filter set 350/420 nm, Ex/Em), one channel to visualize xylenol orange formation labels (TRITC, 545/620 nm) and one channel to visualize calcein formation labels (FITC, 470/525 nm). The differences in excitation/emission wavelengths ensure minimal overlap of fluorescent signal among images. Each raw three-dimensional image of a vertebra ($6 \times 4 \times 5$ mm in size) was acquired over 5 days (fully automated acquisition). Images of each specimen consisted of 1000 cross-sections using a 4×3 array of image tiles (a total of 36,000 individual images per specimen). The raw images occupied 98 GB of memory per fluorescent channel (294 GB total per specimen). Raw images were transferred to a workstation for processing.

5.3.2 Image Processing and Validation

Image processing to segment bone autofluorescence involved the following steps [181]: automatic tiling to create a mosaic of the images from each cross-section; vertical alignment of individual cross-sections using a fiduciary marker; subtraction of out of plane signal; and segmentation of bone using a local iterative thresholding algorithm. We have shown that this image processing approach allows for visualization of small bone

surface features enabling tracing of individual resorption cavities (identified as eroded surfaces) [182]. Due to the large size of the image data, image processing required 5 days of computational time per specimen (Dell Precision T5400, 32GB RAM, 2.66GHz processor with 1333 MHz front side bus). The fully processed image of the bone in each rat vertebra was reduced to approximately 9 GB of segmented data in binary (single bit per voxel) format.

Xylenol orange and calcein image data were stitched together using positioning information from the bone channel and processed as follows: Serial block face imaging is subject to fluorescence signal originating from below the surface. Out-of-plane signal blurs the edges of features in raw images, thereby exaggerating feature size upon segmentation. Out-of-plane signal was removed by subtracting the attenuated and convolved underlying image (the so-called “Next Image” method) [181, 190]. Serial block face imaging is also susceptible to cutting marks left on the newly revealed specimen block face. Grayscale morphological closing with a disk shaped structuring element was applied to fill gaps in pixel intensity caused by cutting marks. Each image volume was then resampled to produce approximately isotropic voxels ($1.4 \times 1.4 \times 1.7 \mu\text{m} / \text{voxel}$). Segmentation was achieved using manually determined global thresholds for each fluorescence channel. Subsequently, three-dimensional binary morphological closing with a spherical structuring element ($5.6 \mu\text{m}$ in radius) was applied to reconnect labels that were discontinuous due to cutting artifacts. Fluorescence signal from within the marrow space was removed by applying a mask of the thresholded bone volume. The resulting image displayed formation labels within bone as well as a number of small noise specks. Objects less than $10,000 \mu\text{m}^3$ (the volume of ~ 14 osteoblasts, much smaller

than the 2,300 osteoblasts that make up one formation site [65, 143]) were classified as noise and were removed from the image, providing a fully processed image of the formation labels (Figure 5-1c-d). Image processing and visualization were performed using custom software written with Matlab (R2008a with Image Processing Toolbox, Natick, MA, USA) and scripts written for Amira (5.3 Visage Imaging, San Diego, CA, USA).

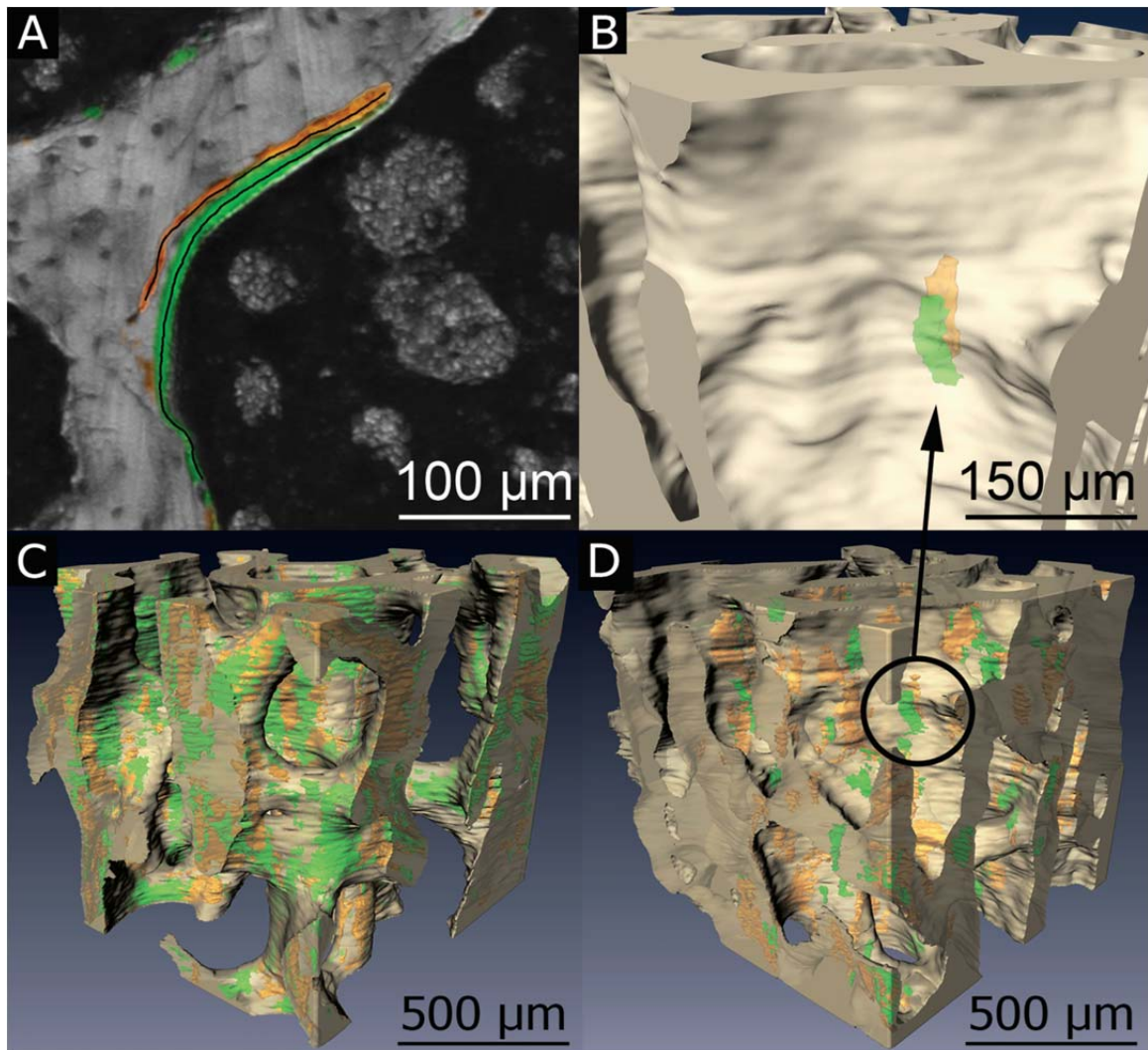


Figure 5-1. (A) Image processing is validated through direct comparison of pixels in the traced labels to corresponding pixels in automatically processed images. (B) A double labeled formation event, consisting of a pair of spatially correlated formation labels (xylenol orange, calcein green), is visible by making the bone surface partially transparent. Images of bone tissue (off-white) and

fluorescent bone formation labels (xylenol orange, calcein green) are shown for **(C)** ovariectomized (OVX) and **(D)** sham operated animals.

Validation of fluorescent bone formation label segmentation was achieved through pixel-by-pixel comparison to manual tracing (Figure 5-1a). Cross-sections of the raw images (5 per specimen) were chosen using systematic random sampling. 'Next-image' processing was applied, but no other pre-processing was performed prior to manual tracing, thereby simulating examination of histology slides. The user then traced each formation label observed in the image using a mouse (mimicking tracing using semi-automated histomorphometry systems). Errors in thresholding were expressed in terms of false positives (Type I error) and false negatives (Type II error) assuming objects traced by the user to be entirely accurate. Type I error was quantified as the cross-sectional area of objects not identified by the observer relative to the cross-sectional area of objects traced by the observer. Type II error was quantified as the ratio of total arc length of tracing lines not contained in the final three-dimensional image relative to the total arc length of all tracing lines created by the observer. Sensitivity was expressed as the arc length of false negatives divided by the sum of arc lengths for true positives and false negatives.

5.3.3 Three-Dimensional Measurements of Bone Remodeling

Resorption cavities were identified by a trained observer as an indentation in the three-dimensional image of the bone surface and confirmed by observing an irregular bone surface texture (the so-called scalloped surface) in the original gray-scale images [182]. All observable cavities were manually traced on the surface of the three-

dimensional image by outlining the edge of each cavity with the lasso tool in the Amira surface editing module (Figure 5-2a,b). Next, measures of resorption cavity depth, volume, and total volume occupied by cavities were determined. A three-dimensional cubic spline was fit to the outer 50 μm of bone surface surrounding each manually traced cavity (spline fits were insensitive to the selection of 50 μm of surrounding bone; altering the distance to 40 μm changed measures of maximum cavity depth by less than 2% on average). The cubic spline surface was generated with Lagrangian end conditions such that the slope of the spline matched that of the three-dimensional bone surface immediately surrounding the resorption cavity. The spline surface spanned the resorption cavity, thereby approximating the 'pre-resorbed' bone surface (Figure 5-2c,d). Maximum cavity depth was determined as the maximum distance between the 'pre-resorbed' bone surface and the eroded surface within the cavity. Mean cavity depth was determined as the mean distance between the 'pre-resorbed' bone surface and the eroded surface within the cavity. Volume per cavity was determined by measuring the volume enclosed by the 'pre-resorbed' bone surface and eroded surface (Figure 5-2e).

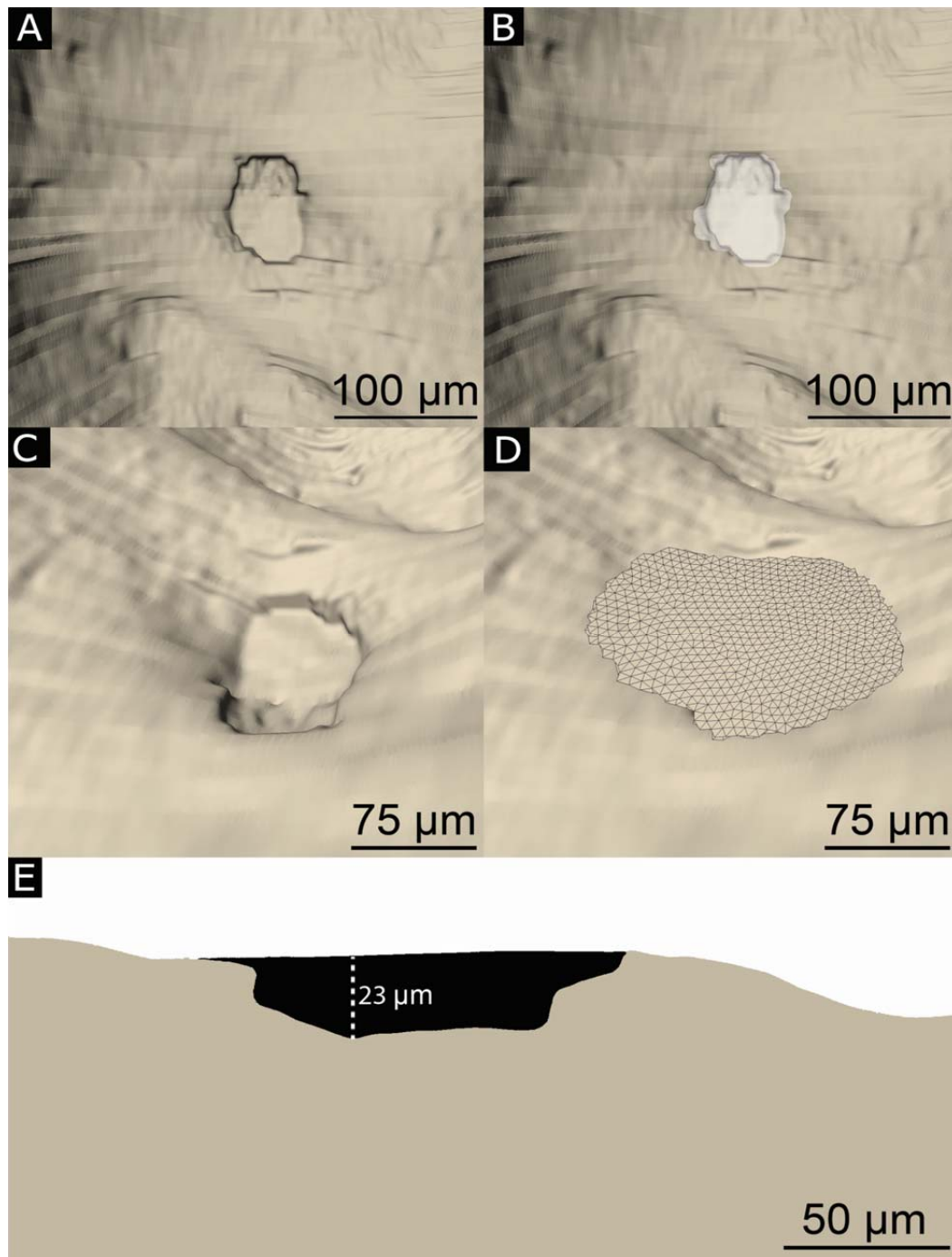


Figure 5-2. (A) Resorption cavities are identified as indentations on the bone surface. The horizontal lines that appear on the reconstructed bone surface correspond to cross-sections during image acquisition. (B) An observer manually traces the resorption cavity (white). (C,D) Maximum Cavity depth is determined by approximating the ‘pre-resorbed’ bone surface above a resorption cavity using a three-dimensional spline (see Methods for a description of the curve fitting). (E) A cross-section of a resorption cavity with the ‘pre-resorbed’ surface applied is shown. Cavity depth was measured as the maximum distance between the ‘pre-resorbed’ surface and the eroded surface (dashed line).

Three-dimensional measures of mineral apposition rate were achieved by determining the volume of bone between a pair of bone formation labels (one xylenol orange and another calcein), referred to here as a “double labeled formation event” (dL.Ev) (Figure 5-1b, 5-3). An automated method was required to identify all pairs of complimentary formation labels. A limit on the search radius for complimentary labels was imposed to avoid considering two very distant labels as a pair. Double labeled formation events were comprised of xylenol orange and calcein labels within 8 voxels of each other (this distance corresponded to 11.3 μm , the 95th percentile of inter-label distance expected for this labeling strategy [183, 191]). A custom written algorithm was used to ‘flatten’ the formation labels into a one voxel thick surface through the midline of the object, mimicking the label centerlines used for tracing formation labels in traditional dynamic bone histomorphometry [192]. Mineral apposition rate determined in three-dimensional images (3DMAR) was expressed by dividing the volume of bone between the flattened formation surfaces by the formation event’s cross-sectional area and interlabel injection time (μm^3 bone formed / μm^2 event surface area / day, Figure 5-3).

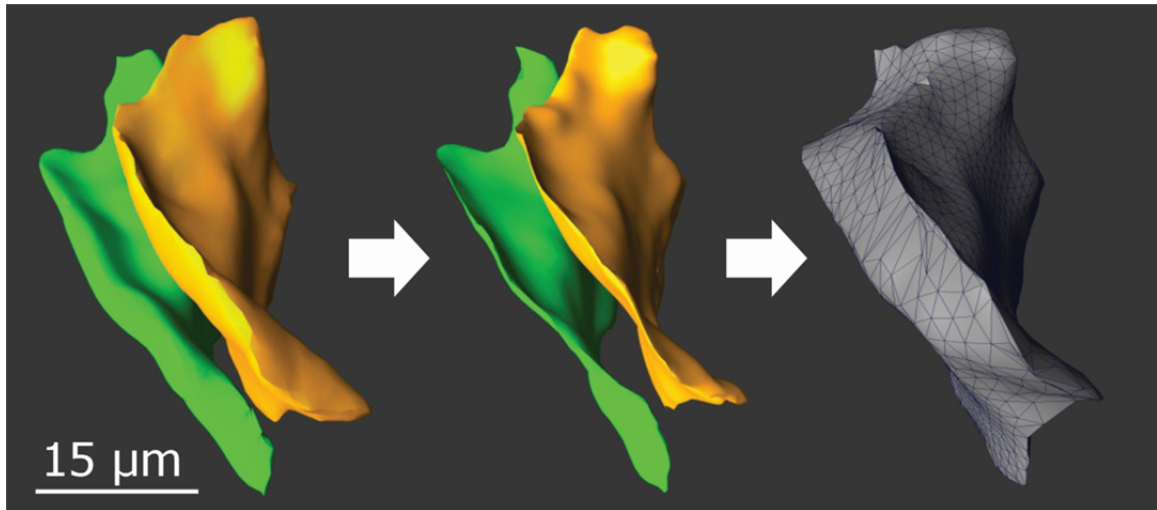


Figure 5-3. A double labeled formation event is identified as a pair of spatially associated formation labels. For quantification, the fluorescent labels are flattened. Mineral apposition rate is calculated as the volume between the flattened labels divided by the label surface area and the inter-label time.

The three-dimensional measures of bone resorption presented here include measures of percent eroded surface (ES/BS), the number of resorption cavities per unit bone surface ($N.Cv/BS$, $1/mm^2$), bone surface area per cavity (BS/Cv , μm^2), maximum resorption depth per cavity ($Cv.De$, μm), mean resorption depth per cavity, volume occupied by each cavity ($Cv.V$, μm^3) and total volume occupied by resorption cavities (presented relative to bone volume, $Tt.Cv.V/BV$). Three-dimensional measures of bone formation include the number of single and double labeled formation events ($N.sL.Ev$, $N.dL.Ev$), surface area per single and double labeled formation event ($MS/sL.Ev$, $MS/dL.Ev$), and mineral apposition rate (3DMAR). Following standard sampling rules, counts of objects (resorption cavities, double labeled formation events, etc.) included only those along three of the six boundaries of the region of interest (in this case, cranial, left, or dorsal boundaries) [128, 193]. All objects present in their entirety (i.e. not in contact with any boundary) were used to measure object surface area, volume or depth

(in the case of cavities). All surfaces (whether in contact with the edge of the sampling volume or not) were included in percent surface area measures (ES/BS, dLS/BS, etc.).

5.3.4 Intra-specimen and Inter-Observer Variation

As with traditional histomorphometry, the location and size of the region of interest examined influence measures of remodeling [193]. To understand the effects of region of interest location and size, three adjacent regions of interest, each 1 mm³ in size, were selected along the cranial-caudal axis of vertebral cancellous bone (Figure 5-1c,d, 5-4, 5-5). In addition to examining measurements made within each 1 mm³, the regions were combined to create 2 mm³ and 3 mm³ regions of interest within each vertebral body (3 mm³ constituted approximately 60% of the entire cancellous bone volume within a vertebra). Differences in three-dimensional measures of resorption cavities, double labeled formation events, and mineral apposition rate among each region of interest were determined using paired t-tests followed by Holm tests for multiple comparisons.

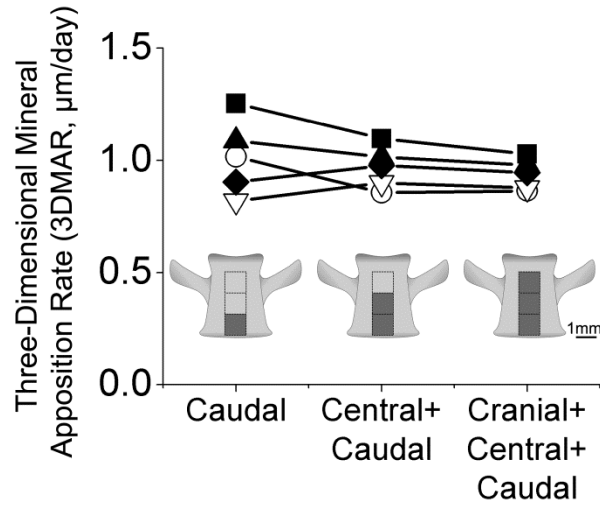


Figure 5-4. Mineral apposition rate in ovariectomized animals (each line is an individual rat) measured in three different regions of interest (caudal, central+caudal, and cranial+central+caudal) are shown illustrating that measurements made in 2 mm³ regions of interest are similar to those achieved using larger regions of interest.

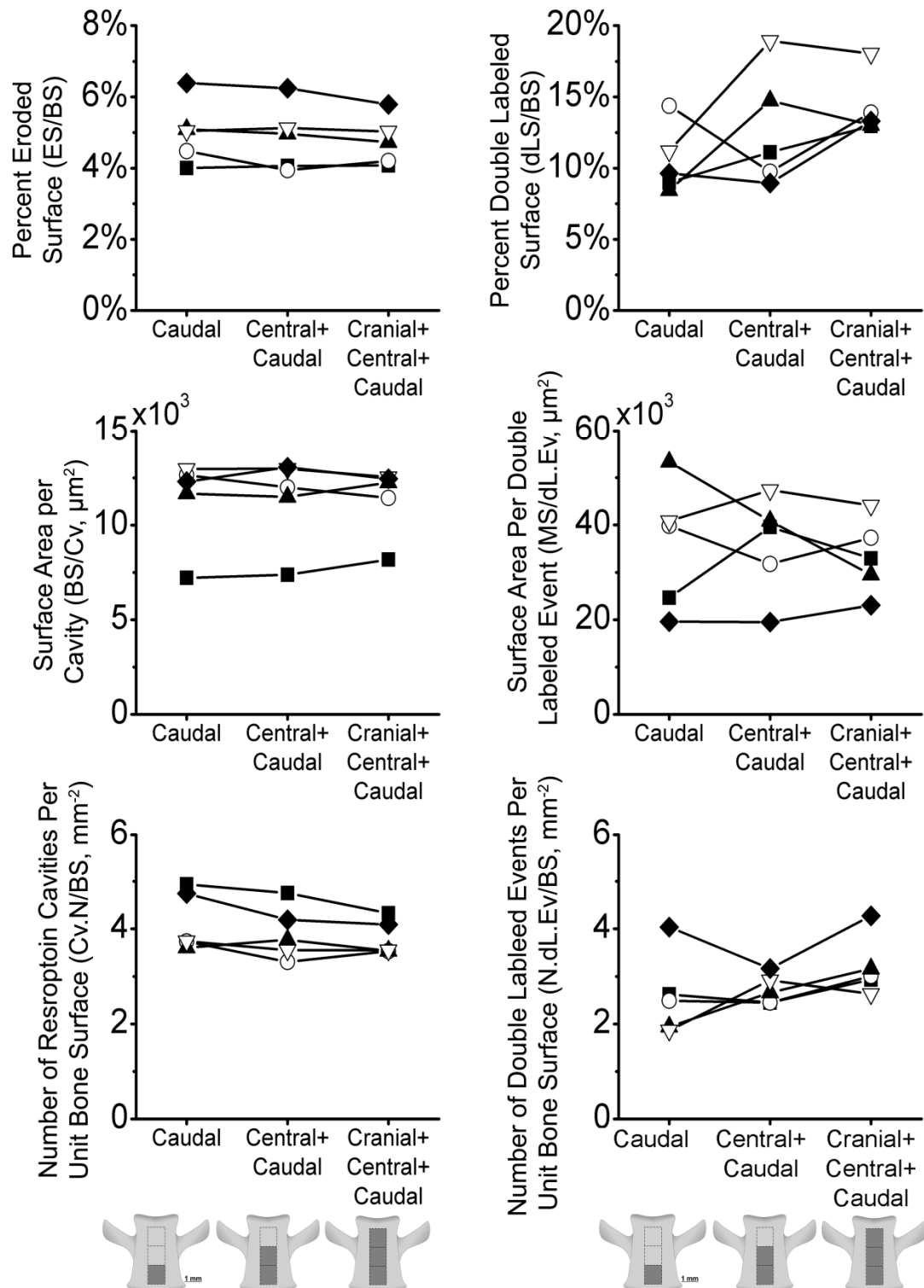


Figure 5-5. The effect of region of interest size on 3D histomorphometry measurements is illustrated. Each line connects data from the same individual. Bone resorption and formation measures achieved in ovariectomized rats are shown for caudal (1 mm^3), central+caudal (2 mm^3), and cranial+central+caudal (3 mm^3) regions of interest. Measurement values converged and variability decreased in larger regions of interest.

Inter-observer variation in measures of resorption cavity surface area and number has previously been established to be less than 10% of the total variance [182]. Inter-observer variation associated with selection of global thresholds for the fluorescent labels was evaluated by separating measured variance in 3D mineral apposition rate and double labeled bone surface into that between observers and among specimens using ANOVA [152, 194]. Differences in cavity and formation label morphology between groups were determined using a generalized least squares model with random effects (REML) to account for multiple measurements per animal (i.e. multiple remodeling events per animal) [195]. Statistical analyses were performed using JMP (8 SAS, Cary, NC, USA).

5.4 Results

5.4.1 Image Processing Validation

There was strong agreement between manual tracing of fluorescent markers of bone formation and the final segmentation achieved using user-defined global thresholding. On average, formation label cross-sectional area in the semi-automated segmentation was 99.8% of that achieved with manual tracing. Sensitivity was $98.6 \pm 0.40\%$ (xylenol orange, mean \pm SD, n=5) and $97.6 \pm 1.64\%$ (calcein, n=5). Measures that can be achieved in two-dimensional histomorphometry (ES/BS, dLS/BS, MAR) were similar to those obtained within the three-dimensional images (Table 5-1). Variance between observers accounted for less than 10% of the total observed variance.

Table 5-1. Measurements achieved in ovariectomized animals (n=5) using traditional two-dimensional techniques are shown along with three-dimensional measures from the same specimens (mean \pm SD). The paired difference between the two approaches is determined within each specimen and the average value across all specimens is shown.

	Two-Dimensional (2D measure)	Three-Dimensional (3D measure)	Paired Difference (2D-3D)/2D*100%
ES / BS (%)	5.01 \pm 0.48%	4.79 \pm 0.66%	-2.33 \pm 8.31%
dLS / BS (%)	10.9 \pm 1.79%	14.2 \pm 2.15%	1.57 \pm 0.19%
MAR (μm / d)	0.92 \pm 0.07	0.94 \pm 0.44	-3.00 \pm 0.06%

When viewed in three-dimensions, individual formation labels have a morphology resembling a “potato chip” (Figure 5-1b, 5-3). Formation labels were situated directly below the bone surface and followed the contours of trabeculae. Formation labels were typically 200 X 200 X 15 μm in size. However, some formation labels exceeded 2 mm in greatest dimension with respect to the x-y-z directions. On average, each specimen contained one or two labels with a dimension greater than 2 mm. Often these outliers intersected the boundaries of the region of interest and could not be included in measures of the formation event size because they were not entirely within the region of interest (but they were included in measures of formation event number and percent surface area, see Methods above). Outliers that were completely within the boundaries of some specimens were not found to bias measures of average formation event size. The amount of xylenol orange formation label (measured as total surface area) and the amount of calcein formation label were similar (within 1.5% of each other by surface area). Both isolated formation labels (single labeled) and paired (a double labeled formation event) were observed. On the order of half of the resorption cavities in each specimen were near

a double labeled or single labeled formation event, as would be expected from the hemi-osteonal theory of bone remodeling. Consistent with the theory of hemi-osteonal remodeling, resorption cavities were also present without neighboring formation labels (presumably a basic multicellular unit that had recently originated). Additionally we observed double labeled or single labeled formation events without a neighboring resorption cavity (presumably a basic multicellular unit that is terminating).

5.4.2 Region of Interest Size

As expected, as the size of the region of interest was increased, the resulting histomorphometry measurements converged. No significant differences in cavity area, 3D mineral apposition rate, or the area of double labeled formation events were observed between 1 mm³, 2 mm³, and 3 mm³ regions of interest (Figure 5-4, 5-5). Measures obtained in 2 mm³ regions of interest were within 12% of the value obtained with a 3 mm³ region of interest (Table 5-2, 5-4, Figure 5-5) suggesting that 2 mm³ is sufficient sampling for three-dimensional dynamic histomorphometry measurements in rat bone. Results from entire 3 mm³ regions of interest are reported to take advantage of all available data.

5.4.3 Alterations in Three-Dimensional Measures of Bone Resorption and Formation Associated with Established Ovariectomy

Established ovariectomy-induced estrogen depletion was associated with a significantly less bone volume fraction and increased percent eroded surface (ES/BS) as

compared to sham operated animals (Table 5-2). Number of cavities per unit bone surface and total volume occupied by resorption cavities were greater in the OVX group (Table 2). No significant differences in average cavity surface area, mean cavity depth, maximum cavity depth, or cavity volume were observed between groups. However, the total volume occupied by resorption cavities was larger in the OVX group ($p < 0.001$). Three-dimensional mineral apposition rate was normally distributed across all formation events and did not differ between OVX and sham groups (Table 5-3). Double labeled bone surface (dLS/BS) was, on average, 23% greater in OVX than in sham specimens ($p < 0.05$, Table 3). The amount of single labeled bone surface (sLS/BS) was small and did not differ between groups. Mineralizing surface as determined traditionally (dLS/BS + $1/2$ sLS/BS) was greater in the OVX group ($15.5 \pm 2.37\%$) than in the sham group ($12.2 \pm 1.41\%$, $p < 0.05$). Cavity size and double labeled formation event size were skewed, but not significantly different from a normal distribution (Shapiro-Wilk test for normality, Figure 5-6). In ovariectomized animals, the number of double labeled formation events per unit bone surface was reduced, but the surface area per double labeled formation event (MS/dL.Ev) was increased ($p < 0.05$).

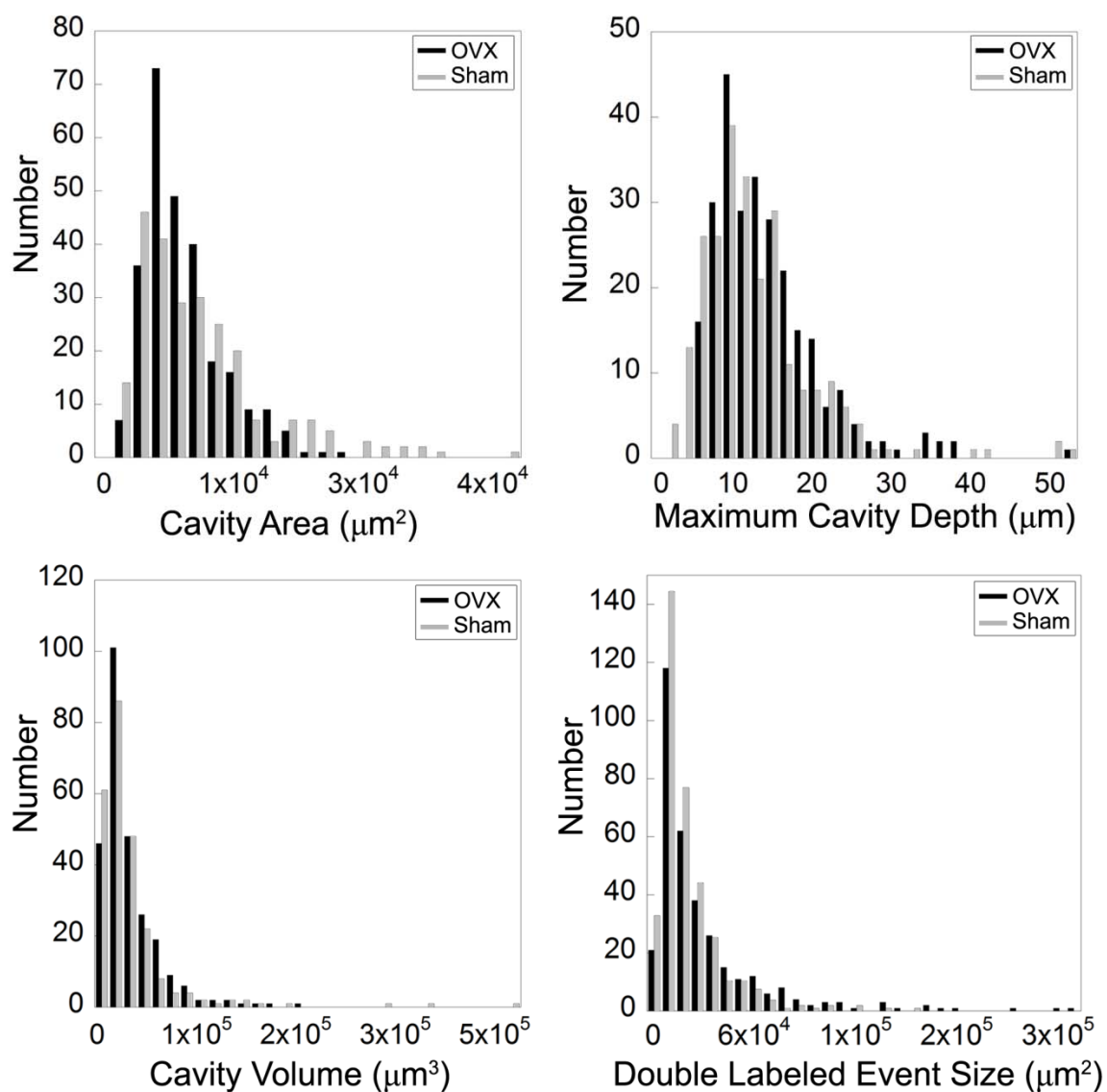


Figure 5-6. Histograms of cavity surface area, maximum cavity depth, cavity volume, and double labeled formation event size are shown. The histograms represent pooled data from all of the specimens examined in the study.

Table 5-2. Static histomorphometry measures made using the three-dimensional approach are shown (mean \pm SD, n=5 per group).

	SHAM	OVX
Bone Volume Fraction	31.5 \pm 7.52%	19.7 \pm 4.0% ^a
Bone Specific Surface (BS/TV, mm ⁻¹)	29.2 \pm 2.20	22.8 \pm 2.17 ^a
Trabecular Thickness (Tb.Th, μ m)	81.3 \pm 8.07	70.5 \pm 3.26 ^a
Trabecular Separation (Tb.Sp, μ m)	221 \pm 41.1	310 \pm 30.4 ^a
Structural Modeling Index (SMI)	0.96 \pm 0.23	1.38 \pm 0.33 ^a
Degree of Anisotropy (DA)	0.68 \pm 0.07	0.68 \pm 0.49
Percent Eroded Surface	2.54 \pm 0.13%	4.79 \pm 0.66% ^b
Number of Resorption Cavities	105 \pm 3.96	134 \pm 13.6 ^b
Number of Resorption Cavities /BS (mm ⁻²)	2.38 \pm 0.24	3.86 \pm 0.35 ^b
Mean Cavity Resorption Depth (μ m)	4.23 \pm 1.85	3.99 \pm 0.88
Maximum Cavity Resorption Depth (Cv.De, μ m)	14.8 \pm 2.56	14.0 \pm 1.96
Average Cavity Surface Area (BS/Cv, μ m ²)	9,720 \pm 1,060	11,200 \pm 1,850
Average Cavity Volume (Cv.V, μ m ³)	36,500 \pm 5,350	52,500 \pm 13,900
Total Volume Occupied by Resorption Cavities (Tt.Cv.V/BV)	0.38 \pm 0.06%	1.12 \pm 0.18% ^c

^ap<0.05, ^bp<0.01, ^cp<0.001 as compared to the sham operated group

Table 5-3. Measures of bone formation made using the 3D approach are shown (mean \pm SD, n=5 per group).

	SHAM	OVX
3D Mineral Apposition Rate (μ m / day)	1.01 \pm 0.46	0.92 \pm 0.44
Average Surface Area per Double Labeled Event (MS/dL.Ev, μ m ²)	23,700 \pm 6,890	33,300 \pm 7,950 ^a
Average Surface Area per Single Labeled Event (MS/sL.Ev, μ m ²)	4920 \pm 4560	5490 \pm 4970
Percent Double Labeled Surface (dLS/BS)	11.5 \pm 0.74%	14.2 \pm 2.15% ^a
Percent Single Labeled Surface (sLS/BS)	3.64 \pm 1.72%	2.52 \pm 0.55%
Number of Double Labeled Formation Events (N.dL.Ev)	152 \pm 41.1	73.8 \pm 20.7 ^a
Number of Single Labeled Formation Events (N.sL.Ev)	187 \pm 89.5	80.0 \pm 20.8
Number of Double Labeled Formation Events Per Unit Bone Surface (N.dL.Ev/BS, mm ⁻²)	5.15 \pm 1.07	3.21 \pm 0.63 ^a
Number of Single Labeled Formation Events Per Unit Bone Surface (N.sL.Ev/BS, mm ⁻²)	6.39 \pm 2.85	3.86 \pm 0.56

^ap<0.05 as compared to the sham operated group

5.5 Discussion

We have demonstrated a new three-dimensional approach to achieving measures of bone remodeling at the level of individual remodeling events. In addition to achieving measures of percent surface area similar to those achieved using traditional two-dimensional histomorphometry, the three-dimensional approach provides a host of new ways to examine bone remodeling including the number and size of individual resorption cavities and bone forming sites (double labeled formation events). Using two-dimensional techniques, it is well known that established ovariectomy in the rat displays a greater proportion of the bone surface undergoing resorption and formation than in estrogen replete animals. The current three-dimensional approach shows that increases in bone resorption and formation in established ovariectomy occur through an increase in number of resorption cavities per unit bone surface with no significant difference in surface area, cavity depth (both maximum depth and mean depth), or volume per resorption cavity. Additionally, established ovariectomy is characterized by a reduction in the number of double labeled formation events per unit bone surface and a corresponding increase in surface area per double labeled formation event.

There are a number of strengths of the current approach that give us confidence in our conclusions. First, the detection of resorption cavities utilized two methods: observation of an indentation on the bone surface in a three-dimensional image followed by confirmation of the scalloped surface in two-dimensional cross-sections (the approach used in conventional histomorphometry) [182]. Additionally, validation of the bone formation label processing was performed through pixel-by-pixel comparison to manual

tracing by a trained observer and resulted in sensitivity in excess of 97%, suggesting that measurements based on fluorescent markers are not biased by the image processing approach. Furthermore, the histomorphometry measures that can be performed in both two- and three-dimensional images were similar and followed the patterns of prior studies of established ovariectomy in similarly aged rats. For example, percent double labeled surface ($14.2 \pm 2.15\%$, OVX in the current study), was within the range of mineralizing surface values reported in prior studies (13.1% - 37.9%) [183, 184, 196]. Three-dimensional measures of mineral apposition rate ($1.01 \pm 0.46 \mu\text{m/day}$) were also within the range of values previously reported using the two-dimensional approach ($0.60 - 2.50 \mu\text{m/day}$) [183, 197-199]. Finally, in traditional dynamic histomorphometry, mineralizing surface is calculated as double labeled surface plus one half single labeled surface. Single labeled surface is included to correct for label escape error when determining mineral apposition rate [200]. The number of single labeled events per unit bone surface was highly variable from specimen to specimen. Such irregularities are reasonable because it is unlikely that all osteoblasts at a remodeling site initiate or terminate matrix formation simultaneously, resulting in an array of small patches of bone formation label. Lastly, while prior studies have reported two-dimensional methods of measuring resorption cavity depth and erosion depth by estimating the pre-resorbed bone surface, the current study is the first to demonstrate such a technique in a three-dimensional image. It is expected that our approach is more accurate than previous two-dimensional approaches as the surface interpolation is in three-dimensions, accounting for cavity morphology out of plane. An additional novelty of our three-dimensional approach is the ability to measure cavity volume and the total volume occupied by resorption cavities.

There are, however, a few limitations to the current study. First, because it is not yet possible to directly view osteoclasts using this imaging modality, resorption cavities were identified based solely on bone surface irregularities (the so-called “scalloped surface”), and are therefore characterized as “eroded surfaces.” Although eroded surface is not a direct measure of active resorption, it is an accepted measure commonly used in histomorphometry. Furthermore, any biomechanical effects of resorption cavities will be present whether or not active resorption is occurring on an eroded surface. Second, the current study addresses only the state of bone remodeling after established ovariectomy in the rat. Prior studies have indicated that in the few weeks following ovariectomy surgery there is a transient increase in bone turnover [187]. It is possible the early reaction to ovariectomy surgery is associated with changes in bone remodeling events that were not observed here. While the current study does not address the early response to ovariectomy, it does address the condition of established estrogen depletion, which is a common pre-treatment condition in animals and humans.

It has long been known that established estrogen depletion would show increases in percent eroded surface as compared to estrogen replete animals. Here we have provided the first quantitative measures of resorption cavity morphology showing that established ovariectomy in the rat is associated with increased number of resorption cavities and total volume occupied by cavities. Only small differences in average cavity surface area (15%), mean cavity depth (6%), and maximum cavity depth (6%) were observed between groups. Post-hoc power analysis suggests that the current study was capable of detecting a 25% difference in average cavity surface area (power = 0.89, α = 0.05), a 35% difference in maximum cavity depth (power = 0.80, α = 0.05), and a 31%

difference in cavity volume (power = 0.83, $\alpha = 0.05$) suggesting that, if these parameters are altered in established ovariectomy, the difference is modest.

Although we observed a decrease in number and increase in size of double labeled formation events in established ovariectomy, we believe that these differences are not caused by alterations in BMU size, but are instead due to formation events from different BMUs merging with one another. There are a number of reasons that we believe the apparent merging of formation events is not an artifact of the image processing methodology. First, the percent eroded and forming surfaces are similar to the literature and the formation label processing is highly sensitive (see above). Second, although morphological processing was used to bridge small patches of bone formation labels for enumeration, our morphological processing approach only bridged patches of fluorescent label separated by 5.6 μm , a distance that is smaller than a single osteoblast. Active osteoblasts within 5.6 μm of one another are likely to experience the same chemical microenvironment, they are effectively functioning as a single formation event [201]. It is likely that merging of formation events was more pronounced in established ovariectomy due to the combined increase in bone turnover and reduction in bone surface. Because the larger and less numerous double labeled formation events in OVX animals appear to be due to merging of formation events from different BMUs, the current study does not support the idea that established ovariectomy is associated with alterations in bone formation beyond that associated with coupling to osteoclastic resorption.

Our three-dimensional approach has the potential to improve our understanding of the connection between in vitro cell culture and in vivo cell activity in the form of BMUs. Basic multicellular units in cancellous bone are believed to form a hemi-osteon,

consisting of active bone resorption with neighboring new bone formation, although visualization of such hemi-osteons is rare in histology sections [10]. In the current study, we observed many remodeling events consisting of a formation surface with a neighboring cavity, consistent with the hemi-osteonal theory of bone remodeling. Additionally, in vitro and in vivo studies suggest that estrogen depletion has been associated with increases in osteoclast precursor number and differentiation as well as decreases in osteoclast apoptosis [64, 177, 186, 202-206]. While increases in osteoclast differentiation presumably lead to an increase in the number of resorption cavities, it has been suggested that reductions in osteoclast apoptosis allow osteoclasts to progress further along or into the bone surface, thereby increasing the surface area or depth of each resorption cavity [207]. Our finding that established ovariectomy is associated with increased number of resorption cavities per unit bone surface is consistent with the idea of increased osteoclast differentiation/proliferation. The fact that resorption cavities are not larger in surface area or depth suggests that if there are alterations in osteoclast apoptosis in this in vivo model of established ovariectomy they do not greatly influence cavity morphology. If our interpretation above is correct and established ovariectomy is not associated with changes in bone formation beyond those secondary to increased osteoclast activity, it would support the idea that the primary effect of prolonged reductions in circulating estrogens on bone remodeling is on the proliferation and differentiation of osteoclast precursors, i.e. the origination rate of new BMUs, and not on BMUs once they are active.

The three-dimensional approach to dynamic histomorphometry described here provides novel improvements over traditional histomorphometry. The degree of

automation in the image acquisition approach greatly reduces labor requirements. Additionally, because measures are performed on digital images, the approach has the potential to reduce intra-laboratory and inter-laboratory variability in remodeling indices. Lastly, the three-dimensional nature of the approach allows for measures of remodeling event number and size. The size and depth of resorption cavities have the potential to influence local tissue stresses, altering the mechanotransduction of local bone cells and potentially influencing failure or failure processes in cancellous bone. Further advancements with this approach include three-dimensional measures of origination frequency (which would be independent of bone formation rate), three-dimensional measures of bone balance, and spatial correlations between new bone formation and trabecular microstructure and/or mechanical stresses and strains.

5.6 Acknowledgements

The authors thank Kathy Tinoco and Ralph O'Brien. This work was supported by NIH/NIAMS R21AR054448, T32 AR007505–22 (C.R.S.), and Cornell University. Dr. Wilson has an interest in BioInVision, Inc., which may commercialize the serial milling technology.

6 Tissue Failure Occurs Preferentially Near Remodeling cavities in Human Vertebral Cancellous Bone

This chapter was submitted for publication in Bone and is currently under review. The article is titled “Microscopic Tissue Damage Forms Preferentially Near Remodeling Cavities in Vertebral Cancellous Bone” by Slyfield, CR, Tkachenko, EV, Fischer, SE, Ehlert, KM, Yi, IH, Jekir, MG, O’Brien, RG, Keaveny, TM, Hernandez, CJ.

6.1 Abstract

The amount of bone turnover in the body has been implicated as a factor that can influence fracture risk and bone strength. A common explanation for the association between bone remodeling and bone fragility is that resorption cavities formed during the remodeling process are stress concentrations that promote local tissue failure and impair whole bone strength. Here we test the idea that cavities promote local tissue failure by determining if microscopic tissue damage (microdamage) caused by controlled loading in vitro is more likely to form near resorption cavities. Specimens of human vertebral cancellous bone (L4, 7 male and 2 female, age 70 ± 10 , mean \pm SD) were loaded in compression to apparent yield (0.8% apparent strain) and stained for microscopic tissue damage using fluorochromes. Three-dimensional images of bone and fluorescent markers of microscopic tissue damage were collected at a resolution of $0.7 \times 0.7 \times 5.0 \mu\text{m}$ using serial milling. The applied loading resulted in a damage volume per bone volume (DV/BV) of $0.77 \pm 0.33\%$. Damage volume per bone volume was correlated with percent

eroded surface ($p < 0.01$, $r^2 = 0.65$), demonstrating that whole specimen measures of bone remodeling and microdamage are related. Individual microdamage sites were $48,700$ ($40,100$; $62,700$) μm^3 in size (median, 25th and 75th quartiles). Microdamage sites had a high surface to volume ratio ($601 \pm 44.5 \text{ mm}^{-1}$), suggesting a crack-like morphology. The proportion of microdamage sites with nearby resorption cavities ($85.3 \pm 7.68\%$) was greater than that of randomly selected bone surfaces without stained microdamage ($36.1 \pm 7.67\%$, $p = 0.001$). Locations within cancellous bone with microdamage were more than two times as likely to have a neighboring resorption cavity than randomly selected sites without microdamage (relative risk 2.39, 95% confidence interval of relative risk: 2.09 – 2.73), demonstrating a local association between microdamage and resorption cavities. That microdamage was associated with resorption cavities when measured at the whole specimen level as well as at the local level provides strong evidence that resorption cavities play a role in mechanical failure processes of cancellous bone. It remains to be determined how microdamage associated with resorption cavities is related to apparent failure properties (toughness, ultimate stress, post-yield deflection, fatigue life) and how variation in cavity number, size and shape due to disease state or drug treatment influence the biomechanical effects of cavities.

6.2 Introduction

Bone remodeling is the primary process of modifying bone structure in adults and is therefore the primary process through which bone loss occurs during aging. Whole body biomarkers of bone remodeling are associated with increased fracture risk

independent of bone mineral density [208], suggesting that, after accounting for bone mass, the amount of bone remodeling in the body may influence bone strength and fragility [209]. Although it is not known how the amount of bone remodeling influences the biomechanical properties of cancellous bone, a commonly cited explanation is that cavities formed during the resorption phase of bone remodeling act as stress risers [178, 179, 210, 211].

To date, evidence that resorption cavities are related to cancellous bone biomechanics has been limited to theoretical and computational models. Computational studies at the scale of individual trabeculae have suggested that the presence of resorption cavities on individual trabeculae causes as much as a 14 fold increase in local tissue stress and strain [15, 212], although, one study did not find the presence of a cavity to greatly influence the mechanical performance of a single loaded trabecula [213]. Studies of cavities on individual trabeculae have been limited to only a few trabeculae (only a total of four trabeculae have been analyzed in this fashion, all from animal specimens) and have not accounted for the complex load distribution within the cancellous bone structure which may over- or under-load trabeculae with resorption cavities. Finite element models of the entire cancellous bone structure have shown that cavities added digitally to the bone surface cause reductions in the stiffness and strength of cancellous bone that are greater than would be expected from the corresponding reduction in bone volume and that the biomechanical effects of cavities are related to cavity location [13, 14]. Theoretical studies therefore support the idea that resorption cavities may influence the biomechanics of the cancellous bone structure and, potentially, that of whole bones.

If resorption cavities do have an adverse effect on cancellous bone biomechanics by acting as stress concentrations, the effect would be detectable at the local level in the form of microscopic cracks or other tissue damage (diffuse damage, trabecular microfracture) occurring in the immediate vicinity of resorption cavities. Traditionally, microscopic tissue damage is studied through examination of two-dimensional microscopy sections. Microscopic tissue damage viewed in this manner is observed in the form of linear microcracks, diffuse damage (diffuse sub-micron scale cracking) and trabecular microfracture [46, 47]. Collectively, these damage morphologies are referred to as microdamage. Microdamage can be generated in cancellous bone by applying mechanical loads in vitro. Previous studies of microdamage measured in two-dimensional sections suggest that local tissue stresses are greater in regions of the structure where microdamage forms [156, 157]. Theoretical models suggest that microscopic tissue damage will form in localized regions of the cancellous bone structure (presumably around stress concentrations) at apparent strains below yield [214]. Experimental studies have shown that reductions in the apparent Young's modulus of cancellous bone can be caused by loading below yield, suggesting that damage and/or microdamage can be generated in cancellous bone at loads below apparent yield [45]. As a result, computational and experimental studies suggest that microdamage may be generated in cancellous bone well before yielding or failure of the cancellous structure.

While theoretical studies support the idea that resorption cavities can influence cancellous bone biomechanics, experimental support for the hypothesis is scant. Scanning electron micrographs of trabecular bone suggest that most resorption cavities are quite broad and shallow and may therefore act as only minor stress concentrations [175].

Furthermore, load sharing within cancellous bone is complex and a trabecula weakened by a cavity may subsequently carry less load, reducing the tissue stress around the cavity. Additionally, it has been proposed that resorption cavities in cancellous bone occur preferentially in unloaded regions of the cancellous bone microstructure, suggesting that such cavities may not have high stresses in surrounding tissue.

We have previously demonstrated an image acquisition and processing technique that can enable visualization of cancellous bone at a scale sufficient to detect individual resorption cavities [181, 182]. Additionally, the same approach has similarly been used to detect and measure microscopic tissue damage labeled with fluorescent markers in bone specimens [215]. In the current study, we combined these two approaches to determine if microscopic tissue damage caused by in vitro loading occurs preferentially near resorption cavities.

The long-term goal of this research is to understand the association between bone remodeling and bone fragility. Specifically, in the current study we determine (1) if microdamage generated by controlled loading in vitro is more likely to be associated with resorption cavities, and (2) whether microdamage assessed in three-dimensional images is correlated with apparent mechanical properties or trabecular microarchitecture.

6.3 Methods

6.3.1 Sample Preparation and Mechanical Testing

The fourth lumbar vertebral bodies of 9 donors (7 male, 2 female, aged 47-80 years, 70 ± 10 , mean \pm SD, tissue source NDRI) were dissected and cleaned of soft tissue. Cylindrical cores of cancellous bone (8 mm in diameter and 15 mm length, one per donor) oriented in the cranial-caudal direction were collected from each vertebral body. Bone marrow was removed with a low pressure water jet. Each specimen was bulk stained in xylenol orange solution (0.5 mM) for 2 hours to label pre-existing microdamage (generated in vivo or during specimen preparation) [133]. The specimens were rinsed in three washes of deionized water, embedded in brass caps using polymethyl methacrylate bone cement [216] and loaded in compression to apparent yield (0.80%) at a rate of 0.5% /sec, after which they were completely unloaded at a rate of 50 N/sec. The gage length for specimens embedded in this manner was taken to be the exposed length of the cylindrical bone core. Elastic modulus, yield strength, yield strain, and residual strain were determined [34]. After unloading, specimens were carefully removed from the testing apparatus and secured in a custom fixture used to prevent subsequent loading on the specimen. The exposed gage length of each specimen was removed using a low speed diamond saw and bulk stained in calcein solution (0.5 mM) for 2 hours to label both pre-existing microdamage and microdamage generated by controlled mechanical loading. The central 5 mm of each specimen was then cut away and embedded

undecalcified in polymethyl-methacrylate made opaque by the addition of sudan black dye in preparation for image acquisition using serial milling [215].

6.3.2 Image Acquisition and Processing

Three-dimensional images of bone and fluorescent labels of microdamage were collected using fully-automated serial milling [181, 215]. Serial milling involves sequentially milling away the top of a specimen and collecting a mosaic of images of the newly exposed specimen block face. Three images of each surface cross-section were obtained: one to visualize bone tissue (using a UV filterset, 350/420 nm, Ex/Em to detect bone by autofluorescence), one to visualize xylenol orange microdamage stain (TRITC filterset, 545/620 nm), and one to visualize calcein microdamage stain (FITC filterset, 470/525 nm). In our implementation, serial milling provided three-dimensional images of bone and fluorescent labels of microdamage at a resolution of 0.7 x 0.7 x 5.0 micrometers per voxel. Raw images of each specimen consisted of 1000 cross-sections, each with a 6 X 6 mosaic. Raw images occupied ~315 GB in memory per fluorescent channel (945 GB total per specimen). Images were transferred to a workstation (Dell Precision T7500, 48GB RAM, 3.47 GHz 64-bit processor) for image processing. Image processing involved the following steps: automatic tiling to assemble a mosaic of each cross-section; vertical registration of each cross-section; removal of fluorescent signal originating from below the specimen block face; and segmentation to identify bone (please see [181] for a thorough description of the image processing steps and validation). The image processing approach resulted in images of bone with sufficient resolution ($< 1.4 \mu\text{m}$) to detect

individual resorption cavities by bone surface texture (identified as eroded surfaces, see [182]).

Images of the microdamage stains were stitched together using positional information determined from the images of bone (method described above). Additional pre-processing steps were then used to segment individual microdamage sites. Images of microdamage were resampled to produce cubic voxels of 2.8 microns in size to reduce computational expense. The outer 0.5 mm of each cylindrical core was removed from each image to avoid including microdamage generated during specimen preparation. Segmentation of each fluorescence channel was achieved using global thresholds identified by a user (Figure 6-1a). After image segmentation, each three-dimensional image was converted to binary format and required 6 GB of memory per specimen. Subsequently, three-dimensional binary morphological closing [170] with a spherical structuring element (5.6 μm in radius) was used to correct for distortions in signal intensity resulting from cutting marks left by the serial milling process. Any fluorescent signal within the marrow space was removed using a masking operation. The resulting images included microdamage as well as small amounts of random noise and non-specific surface staining. Patches of stain less than 10,000 μm^3 (approximately the size of three osteocyte lacuna) were characterized as random noise and removed from the image. Patches smaller than 10,000 μm^3 accounted for less than 1% of the total damage volume. To remove non-specific surface staining from the image, patches of stain that did not extend more than 5 micrometers from the surface were removed. Microdamage caused by controlled loading was differentiated from that naturally occurring or caused by specimen preparation by taking advantage of the pre- and post-mechanical testing stains. The

xylene orange stain identified microdamage that was present before applied mechanical testing while the calcein stain identified pre-existing microdamage as well as microdamage caused by mechanical testing. Sites stained by both xylene orange and calcein therefore represented microdamage present before mechanical loading and were removed from the calcein image, leaving only microdamage generated by controlled mechanical loading (Figure 6-1b,c). For enumeration of microdamage sites, patches of microdamage within 25 μm of one another were considered to be related to one another and were classified as a single microdamage site. Measures of microdamage size and surface to volume ratio were determined for each site. Image processing and visualization were performed using scripts for Matlab (R2008a with Image Processing Toolbox, Natick, MA, USA) and Amira (5.3 Visage Imaging, San Diego, CA, USA).

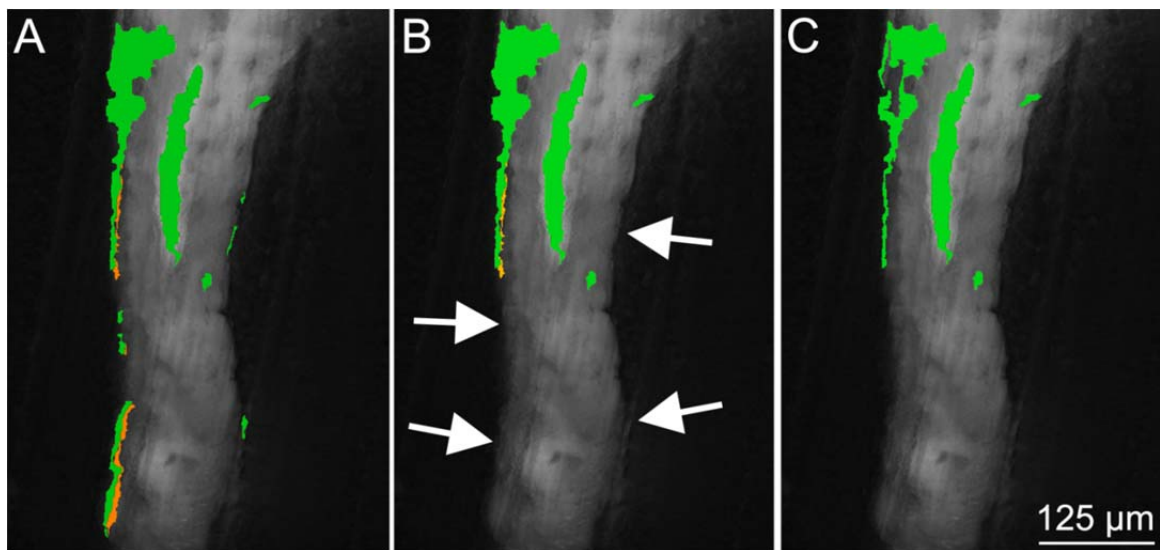


Figure 6-1. (A) A trabecula is shown with xylene orange (staining pre-existing microdamage) and calcein green (staining both pre-existing microdamage and microdamage caused by loading) overlays. (B) Stain that did not extend more than 5 μm into the bone surface was classified as non-specific surface staining and was removed (arrows). (C) Sites with both xylene orange and calcein were removed, leaving only microdamage generated by controlled mechanical loading.

Three-dimensional measures of individual microdamage site volume and surface area to volume ratio as well as the overall damaged volume per bone volume (DV/BV) were determined in each specimen. Measures of volume per site were determined by summing that of all patches within a microdamage site (i.e. all patches of microdamage within 25 μm of each other). Following standard sampling rules, only microdamage sites not in contact with the boundaries of the region of interest were included in measures of microdamage site volume, area, and aspect ratio [128, 170]. Three-dimensional trabecular microarchitecture measures of bone were determined using ImageJ and BoneJ [217, 218]. Microarchitecture measures included bone volume fraction, bone specific surface, trabecular thickness, trabecular separation, structure model index, degree of anisotropy, and connectivity density. Percent eroded surface (ES/BS) was determined by point counting with a grid using three cross-sections per specimen (each 1.25 mm from each other).

6.3.3 Spatial Association of Microdamage and Resorption Cavities

To determine if microdamage sites were more likely to be near resorption cavities, the largest microdamage sites (by volume) were identified (32 sites per specimen). Additionally, locations on the bone surface without neighboring microdamage were selected at random (32 per specimen) (Figure 6-2a). Custom scripts for use with Amira were created to select one of the 64 sites from each specimen in random order and display a three-dimensional image of the 125 μm of surrounding bone (Figure 6-2b).

Grayscale cross-sections of the bone ($0.7\ \mu\text{m}$ pixel size) were also made available to the user making it possible to identify the presence of eroded surface (Figure 6-2c).

Resorption cavities in the region were detected as indentations on the bone surface that also displayed eroded surfaces in cross-section. The observer then recorded whether a cavity was present at the site or not (yes or no). It is important to note that when displayed in this manner, microdamage was not visible so that the user was completely blinded to the presence of microdamage (Figure 6-2b,c).

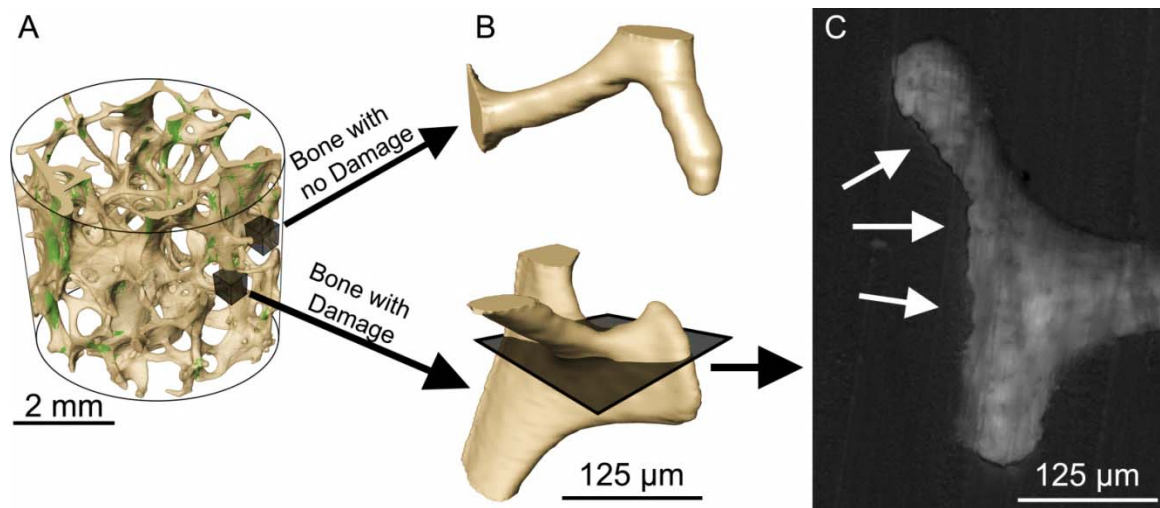


Figure 6-2. (A) The regions of bone with the largest microdamage sites were identified (32 per specimen). Subsequently, regions of bone without microdamage were selected at random (32 per specimen). (B) The bone surface around each region was displayed without showing microdamage so that the observer was blinded to the presence of microdamage. (C) The observer then determined if a resorption cavity was present in the image by observing an indentation on the bone surface associated with eroded surface (arrows).

6.3.4 Statistical Analyses

Correlation analysis was used to identify relationships between mechanical properties (Young's modulus, yield stress, yield strain and residual strain) and measures

of microdamage (DV/BV, median volume per microdamage site, average microdamage site surface to volume ratio, and number of microdamage sites per bone volume). Correlation analysis was also used to identify relationships between measures of trabecular microarchitecture, percent eroded surface, and measures of microdamage. Additionally, correlation analysis was used to identify relationships between measures of trabecular microarchitecture and mechanical properties. The spatial association of resorption cavities and microdamage sites was expressed as the proportion of bone microdamage sites with cavities and the proportion of non-damaged sites with resorption cavities present. The relative risk of observing a cavity near a microdamage site was then determined. Statistical analyses were performed using SPSS (20, IBM, Armonk, NY, USA) and custom software written for use with R (2011, R Foundation for Statistical Computing, Vienna, Austria, published online as a supplementary file).

6.4 Results

Measures of percent eroded surface, damage volume per bone volume, and microdamage site morphology are shown in Table 6-1. Because the volumes of individual microdamage sites were positively skewed (Figure 6-3a) we report median and the 25th and 75th percentiles of these measures (Table 6-1). Microdamage sites generally appeared crack-like by visual inspection (Figure 6-4a,b). Microdamage site surface to volume ratio was $604 \pm 44.5 \text{ mm}^{-1}$ and was normally distributed (Figure 6-3b).

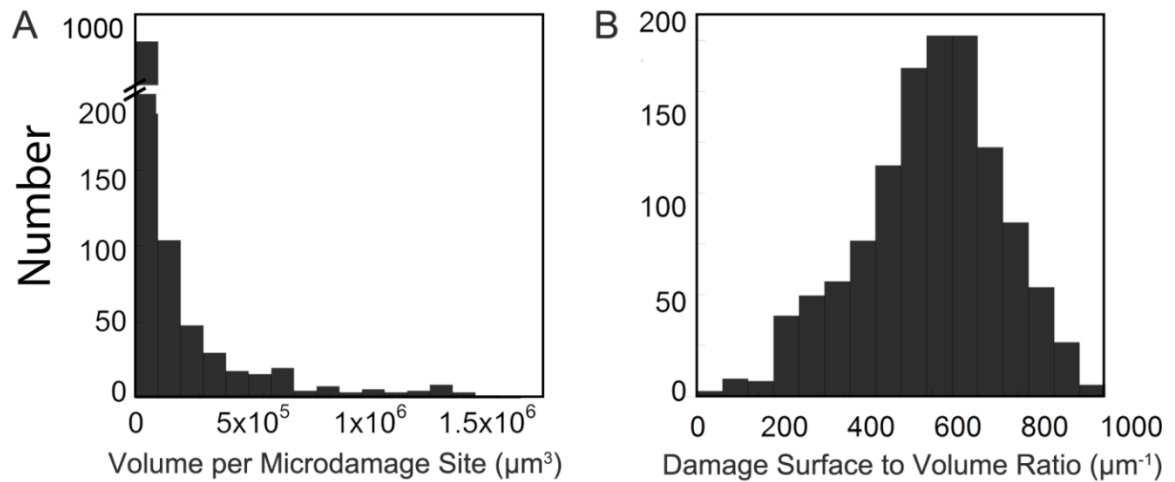


Figure 6-3. (A) The histogram of volume per microdamage site was positively skewed. (B) Microdamage surface to volume ratio was normally distributed.

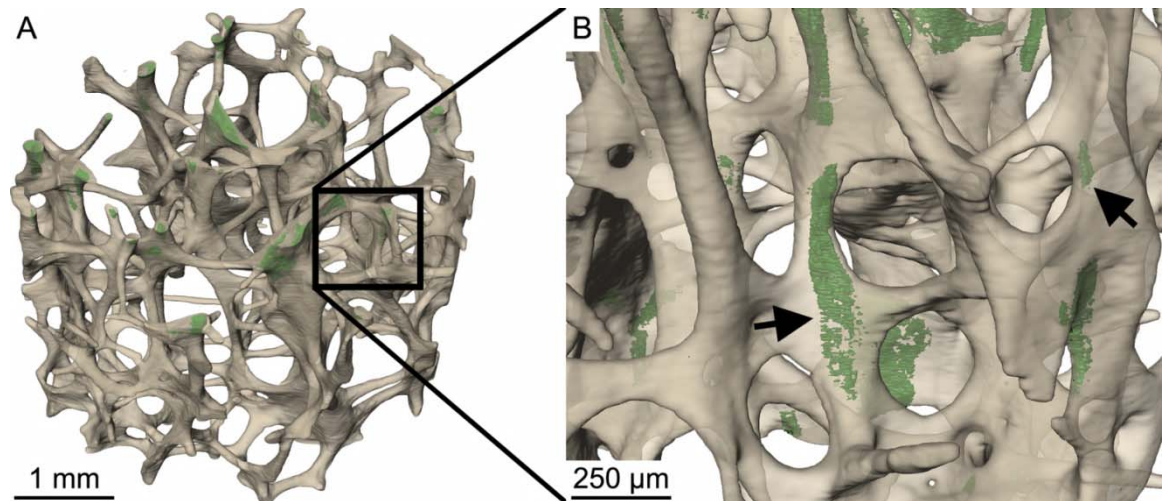


Figure 6-4. (A) The distribution of microdamage generated by controlled mechanical testing (green) within cancellous bone (off-white) is shown in one specimen. (B) Microdamage (arrows) was often diffuse, but tended to be crack-like in that it was thin and broad.

Table 6-1. Static histomorphometry parameters and microdamage measured in three-dimensional images as well as apparent biomechanical properties are shown.

	<i>Mean ± SD</i>
Bone Volume Fraction (BV / TV)	6.75 ± 2.45 %
Percent Eroded Surface (ES / BS)	5.10 ± 1.75 %
Bone Specific Surface (BS/TV, μm^{-1})	1.77 ± 0.50
Trabecular Thickness (Tb.Th, μm)	132 ± 22.2
Trabecular Separation (Tb.Sp, μm)	1,020 ± 217
Degree of Anisotropy (DA)	1.48 ± 0.12
Structure Model Index (SMI)	1.70 ± 0.60
Connectivity Density (Conn.D, mm^{-1})	4.84 ± 2.26
Median Microdamage Site Volume (μm^3)*	48,700 (40,100; 62,700)
Average Microdamage Site Surface to Volume Ratio (μm^{-1})	601 ± 44.5
Number of Microdamage Sites per Bone Volume (mm^{-3})	22.6 ± 10.1
Damage Volume per Bone Volume (DV / BV)	0.77 ± 0.33%
Elastic Modulus (MPa)	232 ± 36.7
Yield Stress (MPa)	1.19 ± 0.26
Yield Strain	0.75 ± 0.04 %
Residual Strain	0.14 ± 0.03 %

*median (25th quartile; 75th quartile)

The proportion of microdamage sites with a neighboring resorption cavity was $85.3 \pm 7.68\%$, while the proportion of non-damaged sites with resorption cavities was $36.1 \pm 7.67\%$ (Figure 6-5, 6-6). The relative risk of a microdamage site having a neighboring resorption cavity was 2.39 (95% CI of relative risk 2.09 - 2.73, $p < 0.001$), demonstrating that microdamage caused by in vitro loading was greater than 2 times more likely to be near a resorption cavity.

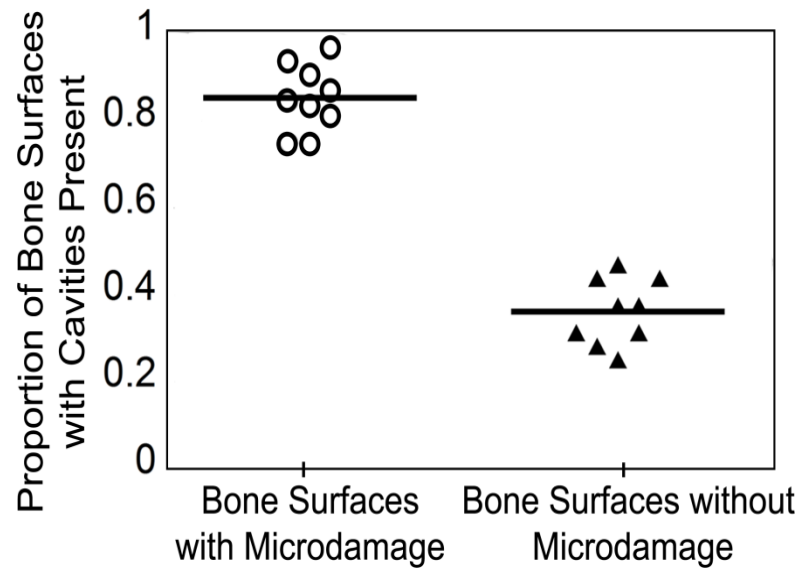


Figure 6-5. The proportion of bone surfaces with cavities was greater in sites with microdamage than in randomly identified locations without microdamage. Locations with microdamage had a relative risk of 2.39 to have a neighboring cavity.

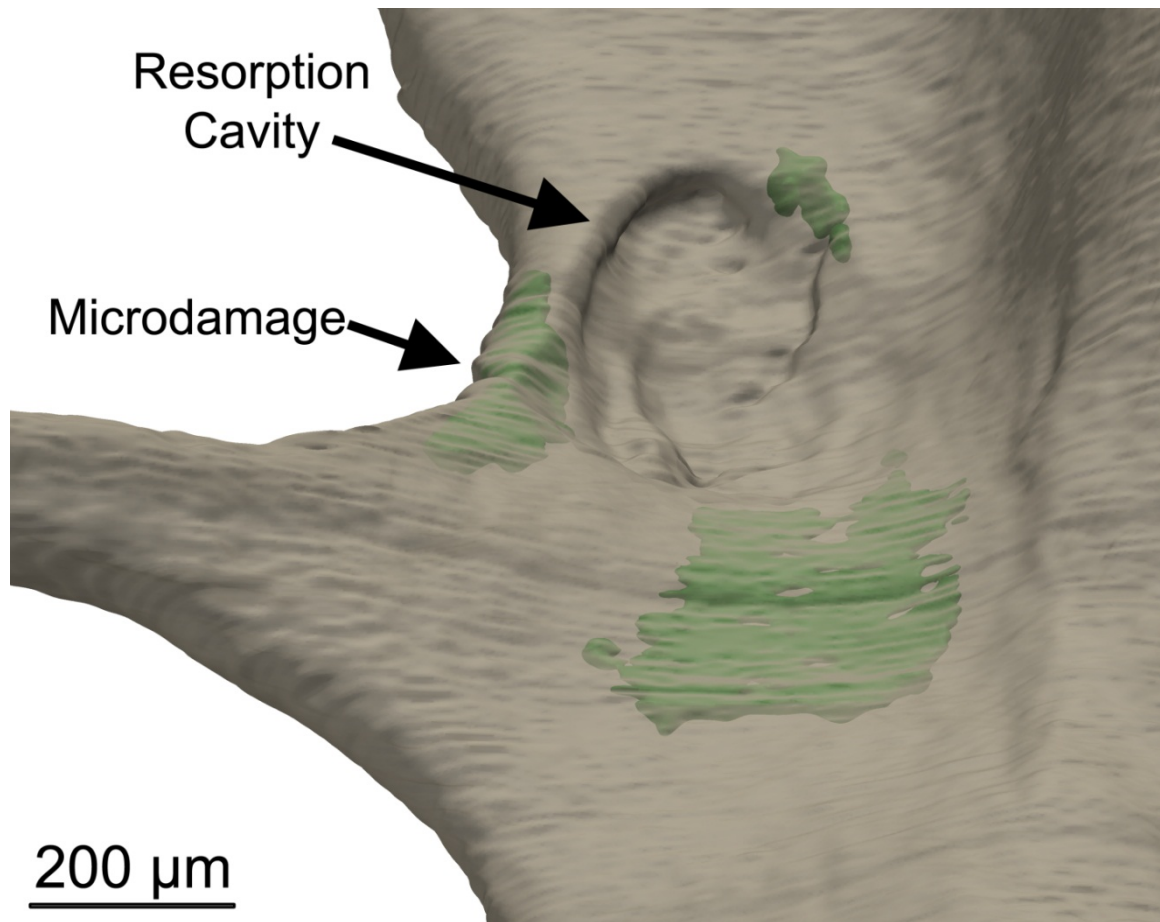


Figure 6-6. Microscopic tissue damage generated by controlled mechanical loading (green) was more likely to be near resorption cavities.

Damage volume per bone volume was positively correlated with percent eroded surface before and after controlling for the effects of bone volume fraction, indicating a relationship between microdamage and the amount of bone resorption in the entire specimen (Figure 6-7, Table 6-2). Percent eroded surface was not correlated with measures of trabecular microarchitecture (Table 6-3). Young's modulus and residual strain were correlated with bone volume fraction, bone specific surface (BS/TV), trabecular separation, and structure model index (Table 6-3). Yield strain was not correlated with any trabecular microarchitecture measure. Yield stress was positively

correlated with bone volume fraction and bone specific surface. Residual strain was correlated with degree of anisotropy after controlling for the effects of bone volume fraction (Table 6-3).

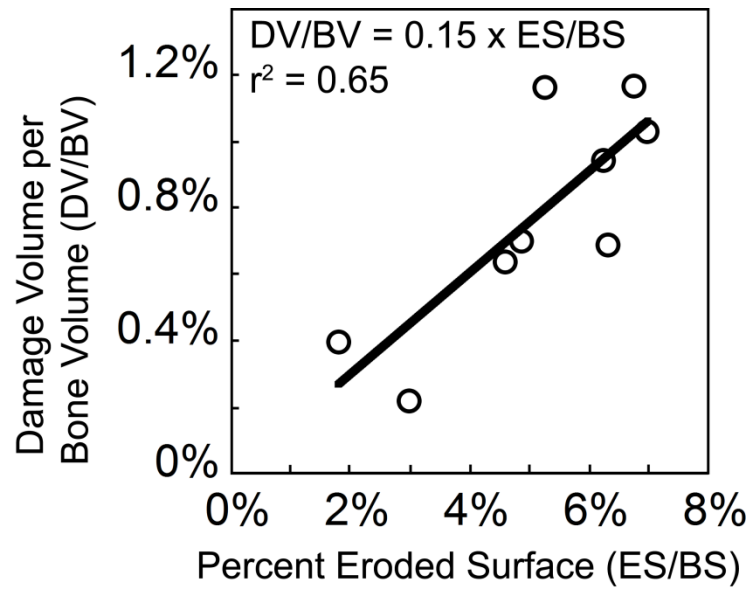


Figure 6-7. Damage volume per bone volume (DV/BV) was positively correlated with percent eroded surface (ES/BS).

Table 6-2. Correlation coefficients between static histomorphometry and microdamage measures (95% confidence intervals). Correlations that are significant do not include zero in the 95% confidence interval.

	DV/BV	Median Volume per Microdamage Site	Average Damage Surface to Volume Ratio	Number of Damage Sites per Bone Volume
ES/BS	0.81 (0.31, 0.96)*	0.13 (-0.58, 0.73)	0.01 (-0.66, 0.67)	-0.30 (-0.81, 0.45)
BV / TV	0.18 (-0.55, 0.75)	0.41 (-0.35, 0.85)	-0.18 (-0.75, 0.55)	-0.48 (-0.87, 0.27)
BS/TV	0.30 (-0.45, 0.81)	0.15 (-0.57, 0.74)	-0.11 (-0.72, 0.60)	-0.18 (-0.75, 0.55)
Tb.Th	-0.14 (-0.73, 0.58)	0.70 (0.08, 0.93)*	-0.23 (-0.77, 0.52)	-0.34 (-0.82, 0.42)
Tb.Sp	-0.07 (-0.70, 0.62)	-0.08 (-0.71, 0.62)	0.01 (-0.66, 0.67)	0.07 (-0.62, 0.70)
SMI	-0.33 (-0.82, 0.42)	0.00 (-0.66, 0.66)	0.12 (-0.59, 0.72)	0.11 (-0.59, 0.72)
Conn.D	-0.05 (-0.69, 0.63)	-0.27 (-0.79, 0.48)	0.10 (-0.61, 0.72)	0.41 (-0.35, 0.84)
DA	-0.02 (-0.67, 0.65)	-0.24 (-0.78, 0.50)	-0.12 (-0.73, 0.59)	0.72 (0.10, 0.94)*

* p < 0.05

Table 6-3. Correlation coefficients between mechanical properties and microdamage measures are listed with 95% confidence intervals in parentheses. No correlations were significant.

	DV/BV	Median Volume per Microdamage Site	Average Microdamage Surface to Volume Ratio	Number of Microdamage Sites per Bone Volume
Young's Modulus	0.11 (-0.58, 0.72)	-0.08 (-0.70, 0.62)	-0.18 (-0.76, 0.55)	-0.19 (-0.76, 0.55)
Yield Stress	-0.03 (-0.68, 0.65)	0.04 (-0.64, 0.69)	-0.05 (-0.69, 0.63)	-0.31 (-0.81, 0.44)
Residual Strain	-0.24 (-0.78, 0.50)	-0.28 (-0.80, 0.48)	0.14 (-0.58, 0.74)	0.17 (-0.55, 0.75)
Yield Strain	-0.28 (-0.80, 0.47)	0.27 (-0.48, 0.79)	0.22 (-0.52, 0.77)	-0.42 (-0.85, 0.34)

6.5 Discussion

The current study demonstrates an association between microdamage generated in human vertebral cancellous bone by controlled loading and the amount of bone remodeling, measured as eroded surface. That the association between microdamage and resorption cavities was present using measurements averaged over entire specimens (DV/BV v. ES/BS) as well as local measures of spatial correlation (relative risk of a cavity being near a microdamage site) provides strong evidence that local failure processes in cancellous bone tissue are influenced by the presence of resorption cavities.

There are a number of strengths to the current study that lend confidence to the results. First, our method of imaging microdamage in three-dimensional images has previously been shown to be consistent with that obtained with conventional two-dimensional analysis [215]. Second, the spatial correlation between resorption cavities and microdamage used techniques designed to reduce observer subjectivity. First, the threshold selection used to detect microdamage was performed in images of microdamage stain alone (bone surfaces and resorption cavities were not visible during threshold selection). Secondly, spatial associations were performed in such a manner that the observer could not simultaneously view both the bone surface and neighboring microdamage. In conventional histology sections, it is not possible for an observer to be completely blinded to neighboring morphology (it is not possible to view resorption cavities in a stained histology section without also seeing neighboring microdamage and vice versa). Additionally, resorption cavities were identified not only as indentations on

the cancellous bone surface, but also through observation of the eroded surface (the standard method of observing eroded surfaces in traditional histomorphometry).

There were a few limitations that must be considered when interpreting our results. First, it is important to note that we have shown that microdamage sites in cancellous bone are more likely to be near resorption cavities. We have not demonstrated that resorption cavities can predict where microdamage is likely to form. Determining the degree to which resorption cavities predict the formation of microdamage is technically more challenging because a fluorescent staining method to allow automated detection of resorption cavities in cancellous bone is currently unavailable. Therefore, manual detection and tracing of all resorption cavities in a specimen would be required (a process expected to take more than 1 week per specimen, that we deem too labor intensive to be feasible). Second, our tests of spatial correlation were limited to the largest microdamage sites in each specimen. Examination of all microdamage sites (~130 sites per specimen) would likewise require excessive labor. However, the largest microdamage sites are also expected to be the most influential with regard to cancellous bone biomechanics. Next, the current study did not identify a correlation between resorption cavities/eroded surface and apparent mechanical properties. One reason is that the current study only examined elastic and yield properties (Young's modulus, yield strain, yield strength, residual strain from loading to apparent yield). The current study was not designed to examine post-yield or failure properties, which are expected to be more sensitive to the formation of microdamage. It remains to be seen if resorption cavities influence post-yield properties of cancellous bone. Additionally, it is possible that microdamage associated with

resorption cavities has a greater effect on cancellous bone submitted to more than one loading cycle (cyclic fatigue or reloading after an isolated overloading event).

The current study adds to an increasing literature examining microdamage in cancellous bone generated in vitro by controlled loading. While the standard method of measuring microdamage is examination of histology sections [130-132], three-dimensional images of microdamage in cancellous bone have been reported using serial milling [215] as well as micro-computed tomography with radiopaque stains (lead uranyl acetate [142] and barium sulfate [141]). The serial milling technique used in the current study has a few advantages over radiopaque staining methods. First, serial milling can be performed with a sub-micron in-plane resolution ($0.7 \times 0.7 \times 5.0 \mu\text{m}$) on an entire cancellous bone specimen (8 mm in diameter). We have previously shown that a voxel size of $1.4 \mu\text{m}$ in-plane or less is required to detect individual resorption cavities [182]. As most micro-computed tomography devices are limited in specimen size as voxel size is reduced (typical maximum overall specimen size is 1 mm for the $\sim 1 \mu\text{m}$ voxel size required to detect resorption cavities), it would not be possible to determine the spatial correlation between microdamage and cavities in entire cancellous bone structures (4-5 mm in smallest dimension) using micro-computed tomography. Additionally, because our technique uses water soluble fluorochromes, we are able to perform sequential staining [131, 133, 135], potentially allowing future studies of crack propagation. A disadvantage of the current approach is the image acquisition time (approximately 1.5 weeks per specimen in the current study). The coefficient of variation (SD/mean) of measures of damage volume per bone volume in the current study (42.8%) was comparable to that reported previously using serial milling (44.7%) [215]. Additionally,

the coefficient of variation in our study (42.8%) is much lower than that reported for two-dimensional techniques or radiopaque staining, which ranged from 53% - 146% (the one exception is a study by Tang et al. 2008 in which variance among specimens was controlled by using tissue from only one donor) [141, 219-221]. High variance can make it challenging to detect differences among study groups [153]. The three staining approaches may also differ in terms of non-specific staining; very little microdamage is detected using pre-stains with fluorochromes or unloaded specimens subjected to lead uranyl acetate [142, 215, 222]. Prior studies using barium sulfate staining have reported amounts of microdamage in unloaded specimens ($1.23 \pm 0.81\%$) that are greater than what we observed in loaded specimens the current study ($0.77 \pm 0.33\%$) [141]. As there has never been a direct comparison of serial milling/fluorochrome staining, lead uranyl acetate and barium sulfate staining, it is unclear if the differences are due to image resolution, signal-to-noise ratio, or the chemistry of the damage stains themselves. Lastly, the current study identified resorption cavities based on indentations on the bone surface and eroded surface and did not determine the presence of active osteoclasts on bone surfaces. However, osteoclast surface is a subset of eroded surface and eroded surfaces within resorption cavities are likely to influence cancellous bone biomechanics whether or not active osteoclasts are present.

In the current study, microdamage sites tended to be crack-like, as indicated by high surface to volume ratios. Others have used the surface to volume ratio of individual microdamage sites to quantify the aspect ratio of individual microdamage sites and classify microdamage sites as crack-like or diffuse [220]. Our measures of microdamage site surface to volume ratio ($601 \pm 44.5 \text{ mm}^{-1}$) were substantially larger than those

reported by others using lead uranyl acetate staining and micro-computed tomography images ($80 \pm 15 \text{ mm}^{-1}$) [220, 221]. There are a number of reasons we believe the surface to volume ratio differed between the two studies. First, our study used a much smaller voxel size ($2.8 \text{ }\mu\text{m}$ voxels for microdamage analysis as compared to $10.5 \text{ }\mu\text{m}$ used with lead uranyl acetate). It is well known that the resolution of an image greatly influences measures of perimeter and surface area. Second, the signal-to-noise ratio of lead-uranyl acetate, used previously, is substantially higher than that of fluorescently labeled microdamage used in the current study. The increase in signal-to-noise ratio combined with a 10 fold larger voxel size could greatly influence the surface morphology of microdamage sites through partial volume effects.

Our findings provide strong support for the idea that resorption cavities affect failure processes in cancellous bone. However, there remain many unanswered questions regarding the ability of resorption cavities to explain the association between the amount of bone remodeling and fracture risk. First, as mentioned above, the current study does not demonstrate a relationship between the amount of resorption cavities and mechanical properties of the cancellous bone structure (“apparent mechanical properties”). Future studies considering multiple loading cycles (fatigue loading or reloading following an isolated overload event), ultimate stress, or toughness are likely to be more sensitive to resorption cavities as they are measures of failure of the cancellous bone structure, while the current study concentrates on elastic and yield properties. Second, the magnitude of the stress concentration caused by a resorption cavity will be influenced by the depth and breadth of the cavity, the thickness of the trabecula upon which the cavity is located and the type of loads applied (tension, bending, etc.). It is not known how resorption cavity

morphology (depth and breadth) varies within or among individuals or if resorption cavity size is influenced by local trabecular thickness, location (rod-like v. plate-like), or orientation (longitudinal versus transverse trabeculae). It has long been proposed that aging and disease states can alter the depth and breadth of resorption cavities, but it is unclear how changes in osteoclast/osteoblast proliferation, differentiation, activity and lifespan regulate cavity size and location. Lastly, the current study has concentrated on the appearance of microdamage in cancellous bone in vitro, but has not addressed the biological repercussions of microdamage. In cortical bone, microdamage generated in vivo has been spatially and temporally associated with new bone resorption [50]. If microdamage is also a stimulus for bone resorption in cancellous bone, microdamage formed near a resorption cavity might promote additional bone resorption, leading to rapid local bone loss in a vicious cycle. It remains to be determined if microdamage stimulates new bone resorption in cancellous bone.

6.6 Acknowledgements

Supported by grants from NIH/NIAMS R01 AR057362 (CJH).

7.1 Conclusions and Future Directions

The serial milling approach to three-dimensional imaging of cancellous bone specimens has been advanced in the current dissertation to quantify site-specific remodeling events and sites of microdamage. Another novelty of serial milling is the ability to colocalize bone microarchitecture with metabolic activity and microdamage. The capabilities of serial milling can be expanded further to image other attributes of a bone specimen including degree of mineralization, cellular activity, and mechanical properties. Colocation of these parameters with microarchitecture and metabolic activity can be achieved simply by attaching devices such as a raman spectroscopy, second harmonic imaging microscope, 2 photon or atomic force microscope to the serial milling device. Additional improvements to the serial milling technology can be achieved by increasing the out-of-plane resolution with the addition of a confocal microscope. Confocal microscopy is capable of achieving sub-micron out-of-plane resolution. The out-of-plane physical dimension of specimens viewed with confocal microscopy is limited to 100 microns. The implementation of confocal microscopy with serial milling overcomes the limitation in physical specimen size. The proposed advancements to the serial milling technology have the potential to bridge the gaps between cell biology, tissue mechanics, and whole bone fragility.

The current dissertation provides a novel method to examine the morphology of bone cell microenvironments at individual remodeling events. From a biological perspective, bone remodeling is characterized by the proliferation, differentiation, vigor, motility, and apoptosis of bone cells. However, bone mechanics is presumably influenced

by the morphology (number and size) of individual remodeling events. The techniques developed in the current dissertation can connect the microenvironment morphology to bone cell biology. Changes in bone cell biology can alter proliferation, differentiation, vigor, motility, and apoptosis. Bone cell functions modify the morphology of individual remodeling events. Those changes in remodeling events translate into alterations in bone turnover and explain the biomechanical mechanisms behind metabolic bone disease.

How changes in individual remodeling events influences fracture risk is currently unknown. The three-dimensional imaging techniques and analyses outlined in the current dissertation allow, for the first time, measures of individual remodeling events in cancellous bone specimens. This capability provides a novel method to determine precisely how metabolic bone diseases and clinical drug therapies alter bone fragility and fracture risk. For example, in postmenopausal osteoporosis, increased turnover is correlated with increased fracture risk [11]. The techniques in chapters 4-6 provide a method to understand the relationship between bone remodeling and fracture risk. The serial milling approach provides precise three-dimensional surface texture visualizations of the bone surface. With the image acquisition and processing approaches present in Chapter 4, a subsequent study [182] showed that resorption cavities cannot be visualized with voxel sizes greater than 1.4 μm . Using an animal model of postmenopausal osteoporosis, we have shown that estrogen depletion influences bone remodeling at the level of individual remodeling events. Our finding that estrogen depletion did not change remodeling event size suggests that estrogen depletion does not affect mature osteoclasts or osteoblasts. Our observation that estrogen depletion increased the number of remodeling events suggests an upregulation of bone cell proliferation and differentiation.

The current dissertation also shows that local failure in the form of microscopic tissue damage is related to local and specimen averaged measures of bone resorption. The increase in the number of remodeling events, and thus the number of resorption cavities, in estrogen depletion may reduce the mechanical properties of cancellous bone [13, 14]. If resorption cavities do alter biomechanical properties, the increased number of resorption cavities may explain the relationship between increased bone turnover and increased fracture risk in postmenopausal osteoporosis.

Here, I propose three lines of investigation based on the techniques and discoveries presented in the current dissertation. The future studies include (1) determining the relationships between remodeling events, biomechanics, and bone cell functions (proliferation, differentiation, vigor, motility and apoptosis) in a state of secondary osteoporosis caused by glucocorticoid therapy; (2) determining the spatial relationship between osteocytes and bone remodeling events; and (3) determining how metabolic bone disease impairs the ability of bone to sense and respond to local tissue stresses and microdamage.

The ability to measure individual remodeling events allows for interpretation of biological changes related to bone cell functions. The proliferation, differentiation, vigor, motility, and apoptosis of bone cells are reflected in changes in the number and size of individual remodeling events (Figure 7-1). Osteoclast proliferation and differentiation alter BMU number and size by increasing the number of mature osteoclasts. Increased osteoclast vigor may increase the rate of bone resorption at individual remodeling events by increasing the secretion of protons and enzymes by osteoclasts. Alternatively, increased osteoclast vigor may increase resorption cavity breadth by increasing osteoclast

motility. Osteoclast apoptosis may influence resorption cavity depth by determining how long an osteoclast resorbs bone. Osteoclast apoptosis may also influence resorption cavity breadth by limiting how far an osteoclast moves over the bone surface. Because bone formation occurs preferentially within resorption cavities [10], changes in osteoclast cell function have similar effects on the number and size of formation events. Alterations in osteoblast cell functions also influence the morphology of formation events. Reduced osteoblast proliferation and differentiation can reduce the size of formation events by limiting the number or availability of preosteoblasts and fully differentiated osteoblasts that can form bone. Stromal cells, osteoblasts, and preosteoblasts are required for preosteoclasts to differentiate into fully mature osteoclasts that can resorb bone. An increase in the number of fully differentiated osteoclasts can increase remodeling event number and size. Since osteoblasts are not motile cells, they form bone in the location on the bone surface where they initially attach. Therefore, changes in osteoblast vigor only influence the amount and rate of bone formation. Changes in osteoblast apoptosis can also alter the amount of bone formation by influencing the period of which an osteoblast can form bone.

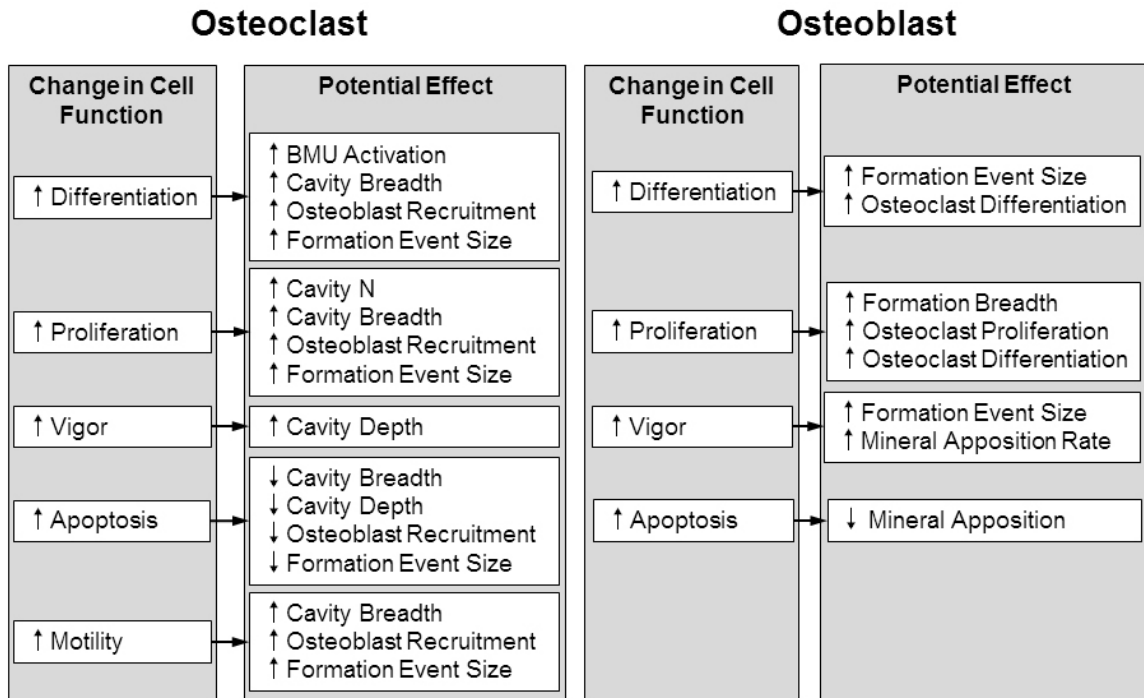


Figure 7-1. Changes in bone cell functions can be measured in terms of the number and size of remodeling events.

The techniques developed here to image and quantify three-dimensional remodeling events can be applied to study how changes in bone cell function ultimately influence bone fragility and fracture risk. Traditionally, measures of resorption and formation were limited to average measures achieved with two-dimensional histomorphometry or whole body biomarkers of bone turnover. The limitation of traditional techniques is important because average measures cannot distinguish between many small resorption cavities or fewer, deeper resorption cavities. The two conditions are quite different biologically and biomechanically. The techniques developed in the current dissertation can be applied to better understand how bone remodeling increases bone fragility in other cases of metabolic bone disease such as secondary osteoporosis. For example, one common cause of secondary osteoporosis is glucocorticoid therapy.

Glucocorticoid therapy are associated with increased fracture risk greater than would be expected from reductions in BMD [223]. Glucocorticoid usage is also associated with increased bone resorption and decreased bone formation, resulting in a net reduction in bone turnover [75]. While it is believed that the increased fracture risk with glucocorticoid use is due to changes in bone cell functions [75], precisely how the changes in bone cell function influence bone turnover and bone biomechanics is not known. Changes in bone turnover are determined by the number and size of individual remodeling events. The number and size of remodeling events can now be characterized in glucocorticoid induced osteoporosis. The changes in remodeling event number and size may influence cancellous bone biomechanics which are responsible for the observed increases bone fragility and fracture risk (Figure 7-2).

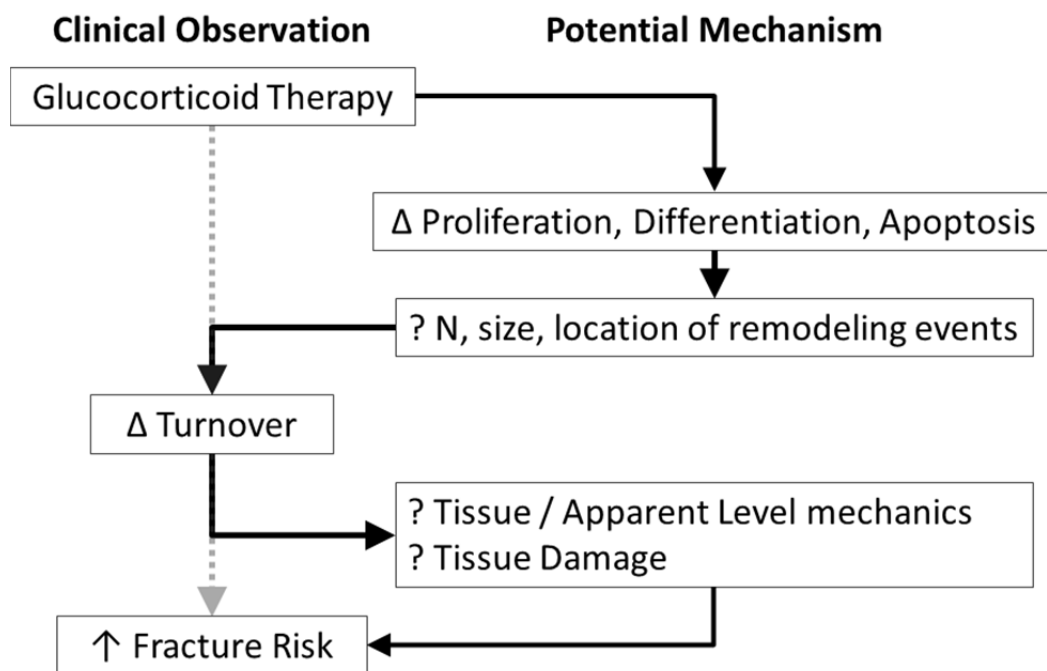


Figure 7-2. The techniques presented in this dissertation can be applied to understand the mechanisms behind changes in bone turnover and increased bone fragility and fracture risk.

The three-dimensional methods of visualizing bone resorption and formation demonstrated in the current dissertation have the potential to enable studies of the function of osteocytes in cancellous bone. Osteocytes are believed to act as the mechanical sensory cells in bone. Osteocytes have been proposed to sense changes in tissue stress and strain. Osteocytes locally regulate bone modeling and remodeling to adapt trabecular microarchitecture to changes in load by communicating with each other and with other cells in the marrow space [74, 77, 224-231]. Age, estrogen depletion, and metabolic bone disease may reduce osteocyte number and viability [77, 232-234]. Reductions in the viability and spatial density of osteocytes may impair the ability of bone to sense and adapt to tissue stresses and microdamage. However, to date, no study has been performed that demonstrates a spatial relationship between osteocytes and bone remodeling in cancellous bone. Osteocyte lacunae are visible in raw images collected with serial milling (Figure 5-1a), suggesting that it may be possible to measure the location and spatial density of osteocytes. If osteocytes are important in the local regulation of bone remodeling, their number, location, and spatial density in age and metabolic bone disease may influence cancellous bone biomechanics and fracture risk [158].

The techniques of visualizing and quantifying bone formation three-dimensionally described in Chapter 5 could be applied to understand how impairment of mechanotransduction in metabolic bone disease states can lead to increased fracture risk. The ability of osteocytes to sense and respond to mechanical stimuli may be severely reduced in cases of secondary osteoporosis caused by glucocorticoid therapy. Reduced bone formation as well as increased osteoblast and osteocyte apoptosis have been

observed in glucocorticoid induced osteoporosis [75, 235]. Together, reduced bone formation and osteocyte viability may impair the bone's ability to sense and respond to local tissue stresses and microdamage and may explain the increased fracture risk associated with glucocorticoid therapy [223]. Studying the relationships between bone remodeling and local tissue stresses using an animal model of glucocorticoid induced osteoporosis would improve our understanding of how interruptions to mechanotransduction pathways increase fracture risk during glucocorticoid therapy. A study could be performed where an in vivo animal loading model [236] is used to induce local tissue stresses in cancellous bone. Double formation labeling could be applied in vivo before and after subjecting the animal to loading bouts. The loaded specimen can then be excised from the animal and bulk stained for microdamage generated by the in vivo mechanical loading. Serial milling can then be applied to generate image-based finite element models of the loaded specimens. Subsequently, tissue strains could be determined in regions where bone is being formed or resorbed. The spatial relationship between bone cell activity, osteocytes, microdamage, and tissue strains would determine whether bone cell activity is influenced by local mechanical stimuli in cancellous bone. With control animals as a base point, the spatial relationships between bone formation, resorption, microdamage, osteocytes, and tissue strains would determine precisely how glucocorticoids impair the ability of bone to respond to mechanical stimuli and microdamage. The response of bone remodeling to local tissue stresses would improve our understanding of the regulation of bone remodeling in the development and treatment primary and secondary osteoporosis caused by glucocorticoids and other factors.

The biomechanical mechanisms determining fracture risk are still not understood in many metabolic bone diseases. The current work provides a means to determine how changes in bone biology resulting from different metabolic bone diseases affect bone biomechanics and ultimately fracture risk.

APPENDIX

Table of Contents

A.1.	Voxel size and measures of individual resorption cavities in three-dimensional images of cancellous bone	128
A.2.	Code	149
A.3	References.....	150

A.1. Voxel size and measures of individual resorption cavities in three-dimensional images of cancellous bone

Reprinted from Bone Journal, 45 / 2009, EV Tkachenko, CR Slyfield, RE Tomlinson, JR Daggett, DL Wilson, CJ Hernandez, Voxel size and measures of individual resorption cavities in three-dimensional images of cancellous bone, Pages No., Copyright (2009), with permission from Elsevier

A.1.1 Abstract

Cavities formed by osteoclasts on the surface of cancellous bone during bone remodeling (resorption cavities) are believed to act as stress risers and impair cancellous bone strength and stiffness. Although resorption cavities are readily detected as eroded surfaces in histology sections, identification of resorption cavities in three-dimensional images of cancellous bone has been rare. Here we use sub-micrometer resolution images of rat lumbar vertebral cancellous bone obtained through serial milling ($n=5$) to determine how measures of the number and surface area of resorption cavities are influenced by image resolution. Three-dimensional images of a 1 mm cube of cancellous bone were collected at $0.7 \times 0.7 \times 5.0 \mu\text{m}/\text{voxel}$ using fluorescence based serial milling and uniformly coarsened to four other resolutions ranging from $1.4 \times 1.4 \times 5.0$ to $11.2 \times 11.2 \times 10 \mu\text{m}/\text{voxel}$. Cavities were identified in the three-dimensional image as an indentation on the cancellous bone surface and were confirmed as eroded surfaces by viewing twodimensional cross-sections (mimicking histology techniques). The number of cavities observed in the $0.7 \times 0.7 \times 5.0 \mu\text{m}/\text{voxel}$ images (22.0 ± 1.43 , mean \pm SD) was not significantly different from that in the $1.4 \times 1.4 \times 5.0 \mu\text{m}/\text{voxel}$ images (19.2 ± 2.59) and an average of 79% of the cavities observed at both of these resolutions were coincident. However, at lower resolutions, cavity detection was confounded by low sensitivity ($\leq 20\%$) and high false positive rates ($\geq 40\%$). Our results demonstrate that when image voxel size exceeds $1.4 \times 1.4 \times 5.0 \mu\text{m}/\text{voxel}$ identification

of resorption cavities by bone surface morphology is highly inaccurate. Experimental and computational studies of resorption cavities in three-dimensional images of cancellous bone may therefore require images to be collected at resolutions of 1.4 $\mu\text{m}/\text{pixel}$ in-plane or better to ensure consistent identification of resorption cavities.

A.1.2 Introduction

The amount of bone remodeling in the skeleton has been shown to contribute to the prediction of fracture incidence, independent of bone mineral density [208]. These findings and others have led to the suggestion that bone remodeling can influence the apparent mechanical properties of cancellous bone independent of bone mass [209, 211]. Although there are many potential explanations for how bone remodeling might have a disproportionate effect on bone biomechanics, the most frequently cited explanation is that cavities formed on the surface of cancellous bone during the remodeling process (resorption cavities) act as stress risers and promote bone fragility [179].

Finite element models support the idea that resorption cavities can greatly increase local tissue strains in cancellous bone. Finite element models in which resorption cavities are added digitally to individual trabeculae [212] and whole regions of cancellous bone [13, 14] suggest that resorption cavities cause local stress concentrations and may alter cancellous bone strength and stiffness. A limitation of these theoretical studies, however, is that cavities are added digitally and are of artificial shape, size and location. McNamara and colleagues used finite element models derived from high-resolution ($0.7 \mu\text{m} \times 0.7 \mu\text{m} \times 5.0 \mu\text{m}$) three-dimensional images of individual trabeculae to estimate the stress concentration factors associated with resorption cavities on three different trabeculae [237]. Their analysis suggested that the elastic stress-

concentration factor around cavities ranged from 9.0-14.3 (i.e. stress near the cavity was 9.0-14.3 times larger than the nominal stress applied to the entire trabecula). The work by McNamara and colleagues is unique in that it identified resorption cavities in three-dimensional images of real trabeculae, enabling their biomechanical analysis to consider naturally shaped resorption cavities.

Identification of naturally occurring resorption cavities in three-dimensional images of cancellous bone has been rare due to limitations in imaging modalities. While bone surface irregularities characteristic of resorption, known as the eroded surface (or alternatively the “scalloped surface” or “crenated surface”) can readily be observed in histological sections [129] or by using scanning electron microscopy [175, 238, 239] such images do not lend themselves to three-dimensional measurement in cancellous bone structures or biomechanical analysis. Micro-computed tomography, by far the most commonly used technique to image cancellous bone microarchitecture, has not been used to view resorption cavities at commonly available resolutions (6 μm /voxel or larger). Higher resolution micro-computed tomography (synchrotron or nano-CT) can achieve voxel sizes of 1 μm or less and are capable of observing resorption cavities [240, 241]. The ability to observe resorption cavities in three-dimensional images is therefore likely to be dependent on image resolution, with cavities becoming increasingly difficult to observe in coarser images. It is not known how image resolution influences the ability to detect individual resorption cavities, but understanding the effects of image resolution on the measurement of resorption cavities is necessary to define imaging requirements for future three-dimensional morphological and biomechanical examinations of resorption cavities in cancellous bone structures.

The long-term goal of this line of investigation is to determine the effects of resorption cavities on cancellous bone mechanics. In the current work we determine how image resolution influences visual identification of resorption cavities in three-dimensional images of cancellous bone.

A.1.3 Methods

Six-month old female Sprague-Dawley rats (n=5) from a related study underwent sham surgery by a commercial vendor (Charles River, Wilmington, MA, USA) and euthanized 4 months after surgery (at 10 months of age). Animal use occurred under approval by the Case Western Reserve University Institutional Animal Care and Use Committee. The 4th lumbar vertebrae are dissected free from soft tissue and the posterior elements and endplates are removed using a low-speed diamond saw. Marrow was removed with a low pressure water jet. Specimens are fixed in 10% neutral buffered formalin, dehydrated in increasing concentrations of alcohol and embedded undecalcified in methyl methacrylate made opaque through the addition of Sudan black dye [166].

Three-dimensional images of the vertebrae are collected using a technique called serial block face imaging implemented through fully-automated serial milling. The serial milling approach has been described in detail previously [166]. We refer the reader to the original publication and only briefly describe the technique here. Serial milling is an approach similar to serial grinding in which the upper 5 μm from a specimen is collected and a mosaic of images of the newly revealed surface. The fully-automated process is repeated until the entire specimen has been destructively trimmed away leaving only a high-resolution 3D stack of digital images of the specimen. In our implementation images are collected using epifluorescent microscopy (a UV filter set is used to identify bone based on bone autofluorescence) and a 10X microscope objective is used in the system to achieve a maximum image resolution is 0.7 X 0.7 $\mu\text{m}/\text{pixel}$ in plane and 5 μm out of plane. Because the image resolution is much greater than that obtained by previous researchers (who obtained images at 3.3 $\mu\text{m}/\text{pixel}$ in plane) [166], more complex image processing techniques are necessary to account for camera positioning error, compensate for

fluorescent signal generated by tissue below the imaging plane and iterative thresholding approach for dealing with image noise (we refer the reader to other publications for a detailed description of these methods) [242, 243]. Images obtained through serial milling and processed using these techniques have been shown to allow visualization of resorption cavities on the bone surface [243].

To determine how image resolution influences the ability to detect resorption cavities in three-dimensional images, a sampling volume consisting of a 1mm cube of cancellous bone from the central metaphyseal region of the lumbar vertebra is selected at least 1 mm below the cranial growth plate. The 0.7 X 0.7 X 5.0 $\mu\text{m}/\text{voxel}$ gray-scale images are aligned and stacked into a 3D image and noise caused by out-of-plane signal is removed (as described previously) [243]. The images are uniformly coarsened to 1.4 X 1.4 X 5 $\mu\text{m}/\text{voxel}$, 2.8 X 2.8 X 5 $\mu\text{m}/\text{voxel}$, 5.6 X 5.6 X 5 $\mu\text{m}/\text{voxel}$ and 11.2 X 11.2 X 10 $\mu\text{m}/\text{voxel}$. For clarity we refer to these versions of the images by their in-plane pixel size (0.7 μm , 1.4 μm , 2.8 μm , 5.6 μm and 11.2 μm , respectively). Three-dimensional volume-renderings of each image at each resolution are obtained using Amira (version 4.1.2, Visage Imaging, Carlsbad, CA). A transparent tetrahedral surface rendering was also applied to allow labeling of eroded surfaces while viewing the volume rendering. No smoothing was applied during visualization. To reduce memory requirements during visualization the three-dimensional image is divided into two equally sized overlapping fields of view (9 GB of RAM required for 3D visualization of each field of view at the highest resolution).

Resorption cavities are initially detected as indentations on the cancellous bone surface in the three-dimensional image (Figure A-1). Orthotropic cross-sections of the indentation are then viewed in the gray-scale image. The indentation on the cancellous bone surface is classified as a resorption cavity only if its surface shows irregularities characteristic of osteoclastic resorption, a classification known in the histomorphometry literature as the eroded surface (also referred to as a “scaloped surface” or “crenated surface” in older literature) [129]. Resorption cavities

identified in the specimen are traced manually on the transparent surface rendering using tools available in Amira (the Draw Tool module). In the current study a single observer (EVT) labeled all of the eroded surfaces.

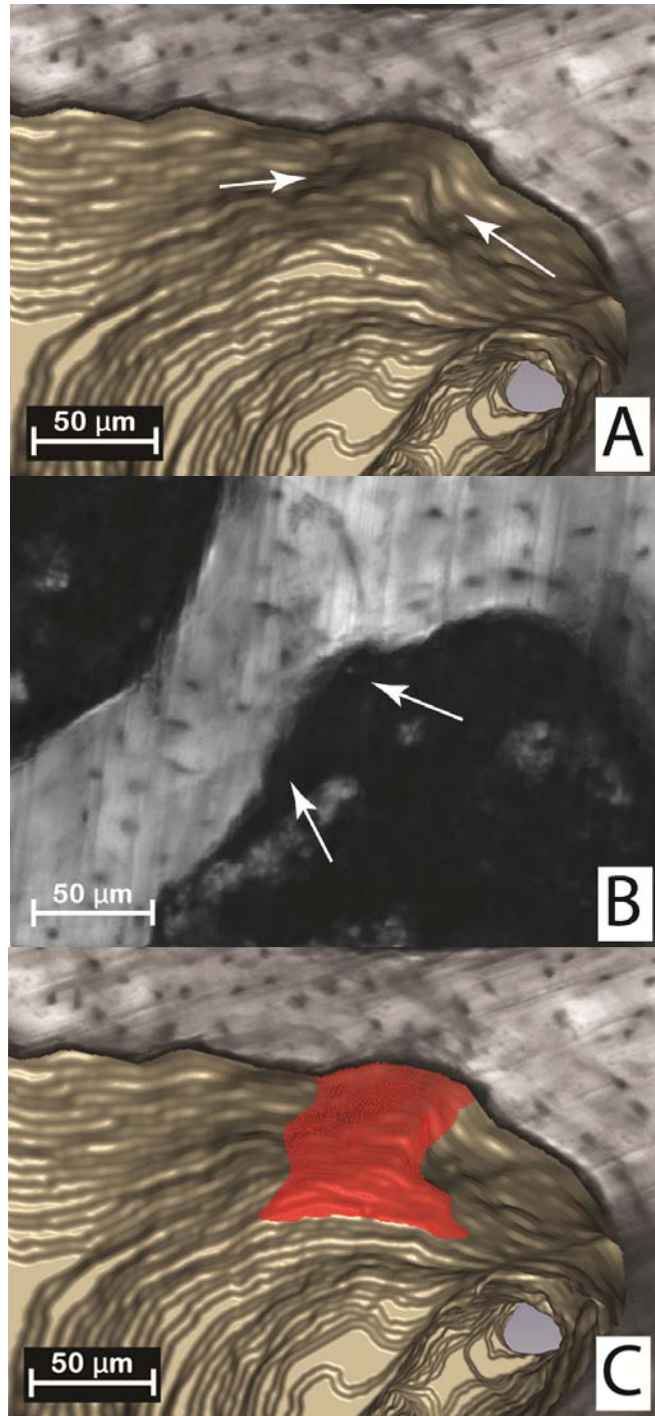


Figure A-1. (A) Initial identification of a resorption cavity on the cancellous bone surface is performed by observing an indentation on the cancellous bone surface in the three-dimensional image (arrows). (B) The presence of a cavity is then confirmed by viewing a two dimensional cross-section of the gray-scale region where the irregular bone surface (arrows). (C) The cavity is then labeled in the 3D image and recorded for future analysis.

The number of resorption cavities, the surface area of each cavity and the ratio of cavity surface to bone surface (ES/BS using histomorphometry nomenclature) are determined for each specimen at each resolution. Following stereology convention, the number of cavities was determined as the number of cavities that are not in contact with a boundary of the sampling volume (in this case a 1mm cube) as well as cavities that are in contact with three contiguous edges of the boundary (caudal, right, or dorsal) [244]. Average cavity surface area is determined from cavities that are entirely within the sampling volume (i.e. not in contact with the boundary of the 1mm cube of cancellous bone). The mean cavity surface area and the distribution of cavity surface area are determined in each image at each of the five resolutions. The only comparable two-dimensional measure of resorption cavities is ES/BS. Measures of ES/BS are collected from two sections selected at random (separated 600 μm from each other) from the 0.7 μm images. The percent eroded surface is determined through point counting at a magnification of 10X and a grid spacing of 0.07 mm.

It is expected that as image voxel size increases (image resolution decreases) detection of resorption cavities will become more difficult, causing the observer to miss some resorption cavities (false negative, Type II error) or erroneously label quiescent bone surface as a resorption cavity (false positive, Type I error). In the current study error is determined relative to measurements made in the highest resolution images (0.7 μm). Bone surfaces labeled as cavities are viewed in a three-dimensional image along with the bone surfaces labeled as cavities in the 0.7 μm image. If a cavity is not labeled in an image but is labeled in the 0.7 μm image it is classified as a false negative. The number of false negative cavities is expressed as a percent of the cavities identified in the 0.7 μm image. If a cavity is present in the lower resolution image but not in the 0.7 μm image it is classified as a false positive. False positives are expressed as a proportion of all of the cavities that are observed in the lower resolution image (i.e. the percent of

cavities that are counted but are not really cavities). Comparisons among image resolutions are performed using paired t-test with Bonferroni multiple comparisons.

A.1.4 Results

Resorption cavities in the highest resolution images (0.7 μm) were readily visible (Figure A-2). As image resolution decreased, resorption cavities became more difficult to observe (Figure A-2). The number of resorption cavities observed declined non-linearly with voxel size (Figure A-3, Table A-1). While no significant differences in the number of cavities were found between the 0.7 μm and 1.4 μm images, a 58% difference in number of cavities was observed between the 1.4 μm and the 2.8 μm images. No resorption cavities were observed in the 11.2 μm images due to the inability to observe eroded surfaces in the two-dimensional cross-sections (Figure A-2). Consistent with the decline in the number of resorption cavities, the percent eroded surface (ES/BS) measured in the three-dimensional images also declined with increasing voxel size. The percent eroded surface measured in the 0.7 μm images in 3D ($2.93 \pm 0.45\%$, mean \pm SD) was not significantly different from the percent eroded surface measured in two-dimensional sections ($2.47 \pm 0.36\%$, $p = 0.07$ in a paired t-test). Small but significant differences in bone volume fraction (BV/TV) and specific surface (BS/TV) were observed among the image resolutions. The mean cavity surface was increased in lower resolution images (Table 1). As voxel size increased cavity surface size showed an increase in mean cavity surface size and a more uniform size distribution (Figure A-4). Errors in labeling resorption cavities greatly increased when voxel size exceeded 1.4 μm . In the 1.4 μm image 22% of the cavities were missed (false negative) and fewer than 10% of cavities that were labeled were characterized as false positive. However, in the 2.8 μm images, as many as 80% of the cavities in the specimen were missed (false negative, type II error) and 42% of the cavities that were identified appeared

as quiescent surfaces in the 0.7 μm images. Greater rates of false positive and false negative errors are observed in images with voxel sizes greater than 2.8 μm .

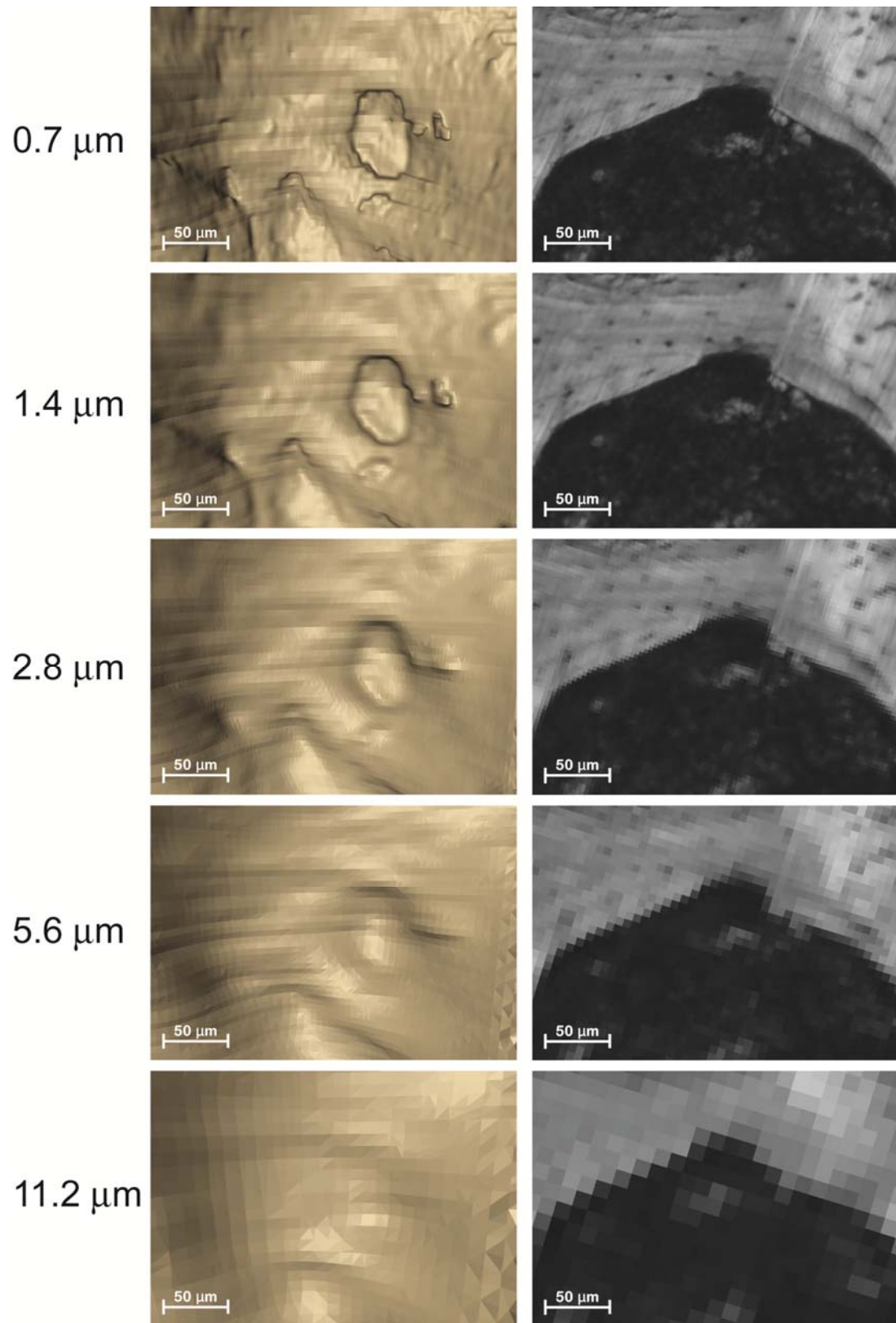


Figure A-2. A region of cancellous bone with a resorption cavity is shown at each of the image resolutions considered in the study. The three-dimensional image is shown on the left and a

single cross-section from the image is shown on the right. As the voxel size increases the resorption cavity becomes increasingly difficult to observe and bone surface irregularities characteristic of resorption cavities become more difficult to distinguish from the bone surfaces. In images with the largest voxel size the eroded surfaces were not discernible from other bone surfaces and no cavities were found.

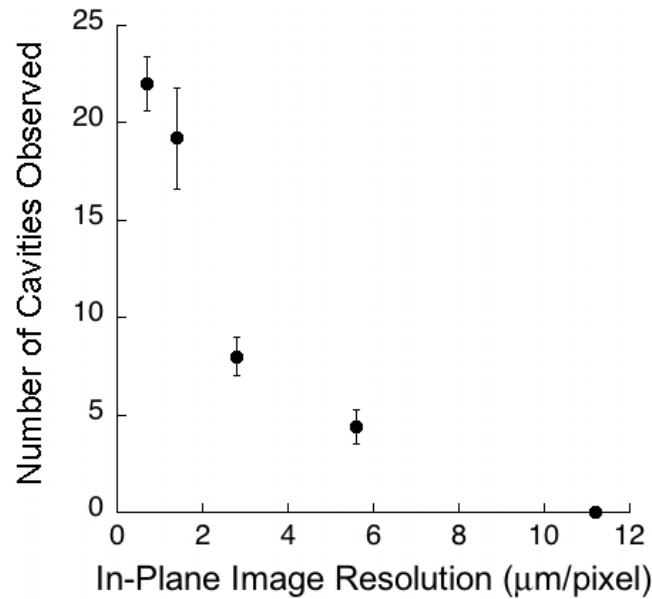


Figure A-3. The relationship between average number of cavities observed in each specimen and the in-plane resolution of the images is shown (out-of-plane resolution is 5 μm/voxel except for the largest images where it is 10μm/voxel). There was an exponential decline in the number of cavities observed as voxel size increased. Error bars represent standard deviations.

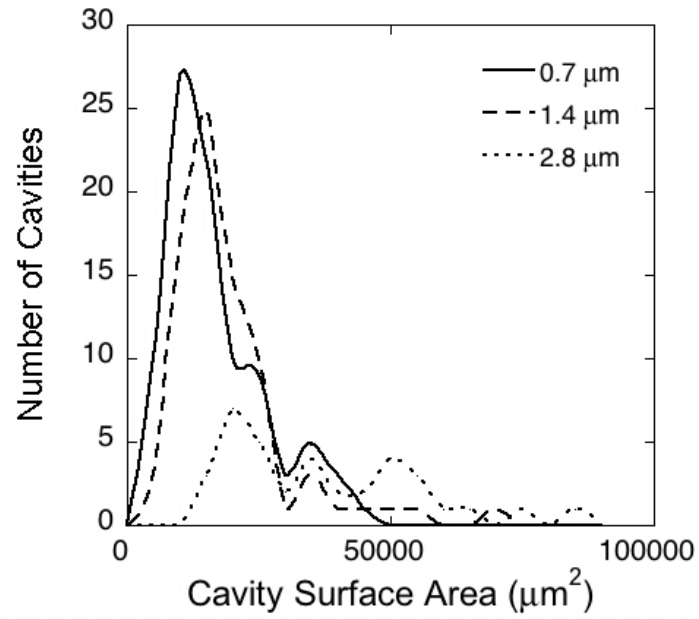


Figure A-4. Histograms showing the distribution of cavity surface size for three different image resolutions (identified by in-plane pixel size). Cavities observed in 0.7 μm and 1.4 μm images showed a similar size distribution, while cavities observed in lower resolution images (2.8 μm shown here) were less numerous and were larger in surface size.

Table A-1. Measurements made from three-dimensional images of rat vertebral cancellous bone were performed at five different image voxel sizes (Mean \pm SD, n=5 specimens). Results with the same superscript letter are not significantly different from one another ($p > 0.05$, paired t-test with Bonferroni multiple comparisons). No cavities were observed in images with the largest voxel size.

	Image Voxel Size				
	0.7 μm	1.4 μm	2.8 μm	5.6 μm	11.2 μm
BV/TV (%)	33.77 \pm 5.80 ^a	34.27 \pm 5.84 ^b	34.86 \pm 5.84 ^c	35.97 \pm 6.02 ^d	38.52 \pm 6.11 ^e
BS/TV (1/mm)	14.92 \pm 2.55 ^a	13.61 \pm 2.49 ^b	12.62 \pm 2.46 ^{a,b,c}	12.13 \pm 2.47 ^c	10.95 \pm 2.23 ^c
ES/BS (%)	2.93 \pm 0.45 ^a	2.74 \pm 0.51 ^{a,b}	2.26 \pm 0.48 ^b	1.01 \pm 0.30 ^c	0.00 ^d
Number of Cavities	22.00 \pm 1.41 ^a	19.20 \pm 2.59 ^a	8.00 \pm 1.00 ^b	4.40 \pm 0.89 ^c	0.00 ^d
Mean Cavity Surface Area ($\mu\text{m}^2 \cdot 10^3$)	14.29 \pm 1.91 ^a	16.47 \pm 4.24 ^{a,c}	34.43 \pm 7.31 ^b	27.02 \pm 4.67 ^{b,c}	NA
False Negatives (% cavities in 0.7 μm images missed in lower resolution)	0.00	21.15 \pm 6.13	80.23 \pm 6.64	88.80 \pm 6.05	100.00
False Positives (% cavities labeled in lower resolution not present in 0.7 μm images)	0.00	9.33 \pm 2.00	42.70 \pm 16.17	43.67 \pm 25.51	NA

A.1.5 Discussion

Based on the current analysis, the ability to detect resorption cavities in three-dimensional images of cancellous bone declines markedly when the voxel size exceeds 1.4 X 1.4 X 5.0 μm /voxel, suggesting that in-plane resolutions of 1.4 μm /pixel in-plane or better are required to reliably detect resorption cavities in three-dimensional images of cancellous bone. One implication of this finding is that biomechanical analyses of resorption cavities may require image resolutions between 0.7 and 1.4 μm in-plane or better to identify the presence of the cavity.

A strength of the current approach is that cavities are detected and confirmed in two ways: first through visualization of an indentation on the cancellous bone surface in a 3D image and second by viewing the eroded surface of the cavity using a two-dimensional cross-section. Previous approaches have identified cavities based on the indentation on the cancellous bone surface [237, 241] or identification of the eroded surface in the two dimensional cross-section (a standard histomorphometry approach) [129] but not both. For this reason we consider the current approach to be more conservative than other approaches to identifying resorption cavities (although it may be less conservative than immunohistochemical methods, see below). We are not aware of previous studies that have quantified resorption cavities in three-dimensional images of cancellous bone and confirmed such identifications using microscopy-based cross-sections. No differences in ES/BS were observed between measures made using the 3D method as compared to the 2D method, strongly supporting the validity of the 3D measurements. Another advantage to the current approach is that it can provide information on the number and surface area of individual resorption cavities in cancellous bone. Traditional, two-dimensional histomorphometry approaches cannot measure the number or size of individual resorption cavities because it is not possible to determine if two isolated eroded surfaces in a two-dimensional section are truly different resorption cavities [155]. Additionally, by examining resorption cavities in a three-dimensional image, the current approach could be used to evaluate the biomechanical effects of cavities in entire cancellous bone specimens using finite element modeling (so far only individual trabeculae have been examined). A final strength of measuring eroded surfaces using our technique is that each labeled cavity is recorded for future examination. This made it possible to directly compare labeled regions at two different resolutions and determine the number of false negative and false positive identifications on a per cavity basis.

There are a number of limitations that must be considered when evaluating the current study, however. First, the images used in the current study had voxels with very large aspect

ratios (the ratio of largest to smallest voxel dimension which equals 7.14 in the 0.7 μm images). The large aspect ratio is due to limitations in the positioning repeatability of the image acquisition device, which constrains the vertical dimension of each voxel to 5 μm at the smallest (see Kazakia et al. for details [166]). The large aspect ratio may influence our ability to detect resorption cavities in the vertical plane of each image since bone surface irregularities are likely to be less visible in that plane. The thickness of each slice of the image, however (5 μm) is similar to that used in traditional histomorphometry, suggesting that specimen thickness is not influencing observations of eroded surfaces when viewed in-plane. However, it remains possible that images with more cubic voxels (aspect ratios closer to 1.0) may be able to reliably detect resorption cavities at a lower resolution than was possible in the current study. Synchrotron- and nano-CT systems commonly achieve aspect ratios near 1.0. Second, identification of resorption cavities in the current study was performed based on bone surface morphology (eroded surfaces) alone and did not specify the presence of active osteoclastic resorption as can be achieved using tartrate-resistant acid phosphatase (TRAP) staining. While morphological identification is sufficient to identify eroded surfaces [129], confirmation of osteoclast presence is required to identify active bone resorption. The implementation of serial block face imaging used in the current study does not have the capability to perform immunohistochemical assays and therefore cannot confirm the presence of active osteoclasts using TRAP staining. However, eroded surfaces (identified based on bone surface morphology) remain a common histomorphometry classification and, it is unclear how the presence of active osteoclasts would influence the biomechanical effects of a resorption cavity. Lastly, evaluation of false positive and false negative cavity detection was performed relative to resorption cavities labeled in the 0.7 μm resolution image, i.e. by considering the measures made in the 0.7 μm images as correct. It is possible that a higher resolution than 0.7 μm could be more accurate with regard to cavity detection. However, because measurements made at 0.7 μm and 1.4 μm are relatively similar (79% of the cavities are coincident at these two resolutions), it is unlikely that reducing the voxel

size below 0.7 μm would lead to very different results, supporting the idea that measurements made at 0.7 μm are an appropriate standard for evaluation of false negative and false positive cavity detection.

Because traditional micro-computed tomography (6.0 $\mu\text{m}/\text{voxel}$) has not been used to view resorption cavities, while higher resolution synchrotron computed tomography or nano-computed tomography ($\leq 1.0 \mu\text{m}/\text{voxel}$) has been used to view resorption cavities, we expected resorption cavities to be increasingly difficult to observe in images with lower resolution [240], which is what was observed. We believe partial volume effects are the primary cause for such errors. In lower resolution images, bone surface irregularities characteristic of an eroded surface may not be visible, preventing the observer from identifying a cavity. At the same time, the discrete nature of the digital image becomes more pronounced in lower resolution images, potentially causing the bone surface to appear irregular when it is not leading to erroneous labeling of quiescent bone surfaces as resorption cavities. The Nyquist-Shannon sampling theorem applied to a two-dimensional image states that to detect a feature in a digital image the feature must be at least 2 pixels in size. While resorption cavities are more than 20 μm deep and can be almost a millimeter in length or width, resorption cavities are not identified by their size but by bone surface irregularities (i.e. the “crenated surface”) [175, 245]. We measured the bone surface irregularities characterized as eroded surfaces in our 0.7 μm images and found such irregularities to commonly be 3-4 μm in length. As a result, it is expected that a pixel size of 1.5-2 μm would be required to identify bone surface irregularities characteristic of eroded surfaces, which may explain the marked increased in errors in detecting resorption cavities as one moves from a resolution of 1.4 μm to 2.8 μm .

Although the current study utilized images obtained through serial block face imaging implemented through serial milling, our results have implications for the use of other image acquisition methods. Imaging modalities such as micro-computed tomography (particularly

synchrotron and nano-CT systems) may achieve cubic voxels of 1.4 μm or less in size and our results would suggest that images obtained with these other modalities are also capable of reliably identifying resorption cavities on cancellous bone surfaces. However, there are two technical limitations that currently limit the use of these other imaging modalities for examination of the biomechanical effects of resorption cavities in cancellous bone. First, current synchrotron and nano-CT approaches show an increasingly limited field of view as image resolution is increased (i.e. only a small region of cancellous bone can be examined at 1 μm resolutions) [246]. Second, many of the existing systems can physically accommodate only very small specimens when using micron or sub-micron scale imaging due to space limitations in the device chambers (often the entire specimen must be less than 3 mm in diameter). Due to these limitations, in their current implementations synchrotron and nano-CT systems may not be able to identify all of the resorption cavities in a continuum level specimen of cancellous bone commonly used in biomechanics studies (3-5mm in smallest dimension [247]). Because the serial milling approach used in the current study is not subject to limitations in specimen size or image field of view [166], it is the most versatile of the three-dimensional imaging approaches currently available for studying resorption cavities in cancellous bone. An additional advantage of the serial milling approach is that it may also be used to image other fluorescent markers such as labels of microscopic tissue damage or bone formation [242].

In their analysis of resorption cavities on individual trabeculae from rats, McNamara and colleagues used an image resolution identical to the highest resolution image in the current study (0.7 X 0.7 X 5 μm /voxel) and referred to the resorption cavities they observed as “presumptive” resorption cavities [237]. Our analysis suggests that resorption cavities are readily visible in images at this resolution and supports the idea that what McNamara and colleagues refer to as “presumptive” resorption cavities may be true resorption cavities, although their detection method was based on observing indentations on the surface of a 3D image, while our approach utilized

both identification of surface indentations in the 3D image as well as observations of bone surface irregularities in the gray scale image (the second step being more consistent with traditional histomorphometry).

The biomechanical importance of resorption cavities remains to be determined. In their examination of three individual trabeculae, McNamara and colleagues found resorption cavities to have a large effect on local tissue stress. In contrast, Eswaran and colleagues used a nano-CT scan (2 $\mu\text{m}/\text{voxel}$) to create a non-linear finite element model of a single canine trabecula displaying a resorption cavity [241]. When the trabecula was loaded in compression the force-displacement curve was similar to that achieved when the same trabecula was modeled at 20 $\mu\text{m}/\text{voxel}$ (although the strength appeared to be slightly decreased in the lower resolution image). It is unclear if the differences in these analyses are due to image resolution (0.7 X 0.7 X 5.0 $\mu\text{m}/\text{voxel}$ by McNamara et al. and 2 $\mu\text{m}/\text{voxel}$ by Eswaran et al.), finite element shape (tetrahedral by McNamara and colleagues v. hexahedral by Eswaran and colleagues), or differences between linear and non-linear modeling approaches (McNamara et al. used a linear modeling approach). As these studies were limited in sample size (together they analyzed a total of 4 individual trabeculae), the biomechanical importance of resorption cavities, if any, remains unclear.

A.1.6 Acknowledgements

This work was supported by NIH/NIAMS AR054448 and the CWRU SOURCE program. The authors thank Justin Daggett, Fred Douglas, Chiderah Okoye and Katherine Ehlert for assistance with handling images.

A.2 Code

Main.m (Human vertebral cores)

```
%% This function creates a tiled mosaic of images of specified grid size.
% The algorithm first calculates the local alignment (alignment
% translation for each pair of adjacent images), and then applies global
% alignment correction (finds minimum set of alignment translations to
% position each image relative to the whole mosaic).

% Abhinav Rao and Evgeniy Tkachenko, March 29, 2011

% This code is used to run specimen: UM07_A024_L4_C4_C
% This specimen was originally processed on BIOMECH-6
% EVT 06.30.11

%% ~~~~~INPUT~~~~~
% raw_image_in      -
% nui_corrected_out -
% tiled_gray_out     -
% tiled_thresh_out   -
% m, n               - 6, 6
% offset_up_bound    - 105

%% ~~~~~OUTPUT~~~~~5

clear all; clc; warning('off');

%% INITIAL PARAMETERS & DIRECTORIES
% General parameters (specimen, # slices)
start_Slice = 1; stop_Slice = 1030;
specimen = 'UM07_A024_L4_C4_C';
% Directories
code_dir = what; code_dir = code_dir.path;
path(path,code_dir)
set_rename = renameSpecimen(specimen);
image_directory = 'G:\Berkeley Specimens\';
save_directory = 'I:\Berkeley Specimens\';
raw_image_dir = [image_directory specimen];
tiled_gray_OP_out = [save_directory specimen '_Processed\Tiled\Gray\OP\'];
tiled_thresh_out = [save_directory specimen '_Processed\Tiled\Thresh\'];
parameters_out = [save_directory specimen '_Processed\Parameters\'];
mkdir(tiled_gray_OP_out); mkdir(tiled_thresh_out);
mkdir(parameters_out);

% Load reference hole slice and shift parameters
RH_Shift = load([code_dir '\RH Location Shift' specimen '_RH_Shift.txt']);
RH_File_Name = [];
for z = 1 : size(RH_Shift,1), RH_File_Name = [RH_File_Name ' ' num2str(RH_Shift(z,1))]; end
```

```

% Open and write logfile
fid = fopen([code_dir '\Processing Logs\Processing_log_' specimen '_' num2str(start_Slice) '-'
num2str(stop_Slice) '.txt'],'w');
fprintf('Currently processing %s: slices %i-%i\n\n',specimen,start_Slice,stop_Slice);
fprintf(fid,'Processing log for specimen %s: slices %i-%i,
%s\n\n',specimen,start_Slice,stop_Slice,datestr(clock));

%% PROCESSING PARAMETERS
% General parameters (grid size, objective size)
m = 6; n = 6; objective = 10;
% Counter for current top and bottom slice number for main loop NUI
top_Num = start_Slice; bot_Num = top_Num + 1;

% SF = scaling factors for OP subtraction
sf_UV = 0.3; sf_CA = 0.4; sf_XO = 0.7;
% Blurring size and extent for OP subtraction
hsize = 5; gspread = 9;
% Global thresholding variables
UVT = 1.25; CAT = .6; XOT = .6;
% Noise removal element area (in pixels)
bwa = 1000;

% Tiles for obtaining global threshold value for ranking tile edges
tiles_Edge_Rank_Thresh = [15, 16, 21, 22];

% Obtain pixel size and reference hole radius parameters...
cd(raw_image_dir);
raw_files = dir(['*' num2str(RH_Shift(1,1)) '_' * '_UV*']);
tile_Size = size(imread(raw_files(1).name));
pixelsize = 3.57*4*(1024/tile_Size(1))/objective;
r = round(.0938*25400/(2*pixelsize));
cd(code_dir);

% Create empty matrices for storing parameters
tile_Offsets_Matrix = struct();
global_X_Coords = struct();
global_Y_Coords = struct();
% Array to store threshold values for each slice
thresh_Val_Array = zeros(stop_Slice,1);

%% 1) Tile Offset & Alignment Acquisition
fprintf('1) Tile Offset & Alignment Acquisition.\n');
fprintf(fid, '1) Tile Offset & Alignment Acquisition.\n');

% Flag for processing all signals (channel_flag=0 if only want UV channel)
channel_flag = 0;

% Obtain tiling offsets and coordinates (loops until user specifies OK)
loop_Flag = 'N';
while loop_Flag == 'N'

```



```

% close all

% Obtain parameters to use for tiling offset reference from user
dlg_title = 'Reference tiling offset parameters';
prompt = {'Reference slice:','NUI kernel size:','NUI reference tile:','SpanT Thresh
Multiplier:',...
'Machine imprecision:','Max machine offset:'};
num_lines = 1;
def = {'135','100','37','1.1','25','105'};
answ = inputdlg(prompt,dlg_title,num_lines,def);

% Slice number for tiling offset reference

slice_Offset_Ref = str2num(answ{1,1});

% Size for averaging kernel for NUI correction

nui_Kernel_Size = str2num(answ{2,1});

% NUI reference tile number (IF a separate NUI tile was not imaged).

% The current setup for the Berkeley specimens is to use the tile number

% 37, as this is the specific tile that was taken to correct the NUI.

% Prior to the Berkeley specimens, no separate tile was imaged, and thus, a

% blank tile in the original grid may be used. For example, tile 1 is used

% for specimen B67959_L4_C3_C.

nui_Ref_Tile = str2num(answ{3,1});

% Threshold multiplier for spanning tree edge weights:

% Increase value to use less weights; decrease value for more weights

spanT_Thresh_Multiplier = str2num(answ{4,1});

% Tiling parameter: imprecision of machine in pixels

imprec_Machine = str2num(answ{5,1});

% Tiling parameter: largest offset observed between two tiles in pixels

offset_Up_Bound_Machine = str2num(answ{6,1});

```

```

% Obtain offsets from user-specified slice

fprintf('\tObtaining offsets from slice %i.\n',slice_Offset_Ref);

fprintf(fid, '\tObtaining offsets from slice %i.\n',slice_Offset_Ref);

fprintf('\t\t-> Loading images & correcting NUI...');

fprintf(fid, '\t\t-> Loading images & correcting NUI...');

tic; [nui_UV_Top,~,~] =
loadImagesNUI(slice_Offset_Ref,nui_Ref_Tile,tile_Size,m,n,raw_image_dir,code_dir,channel_flag,nui_Kernel_Size); t = toc;

fprintf('Finished. Elapsed time: %g seconds.\n',round(t));

fprintf(fid, 'Finished. Elapsed time: %g seconds.\n',round(t));

[tiled_UV_Top,~,~,tile_Offsets,cc_Values,x_Tile_Coords,y_Tile_Coords,spanT_Path,fid] = ...

tileAllChannels(nui_UV_Top,0,0,tiles_Edge_Rank_Thresh,imprec_Machine,offset_Up_Bound_Machine,slice_Offset_Ref,bwa,tile_Size,m,n,fid,channel_flag,spanT_Thresh_Multiplier);

% figure,imshow(tiled_UV_Top,[]); title(num2str(slice_Offset_Ref));

clear nui_UV_Top

% Draw path calculated by spanning tree - DEBUGGING PURPOSES

% fprintf('\t\t-> Drawing spanning tree path...');

% fprintf(fid, '\t\t-> Drawing spanning tree path...');

% tic; path_Mosaic = drawPath(spanT_Path,m,n,tile_Size,10,y_Tile_Coords,x_Tile_Coords); t
= toc;

% resizing_factor = size(tiled_UV_Top,1)/size(path_Mosaic,1);

% overlaid_Path_Mosaic =

imoverlay(mat2gray(tiled_UV_Top),imresize(path_Mosaic,resizing_factor),[1 0 0]);

```

```

% figure,imshow(overlaid_Path_Mosaic*2,[]);

% title(['Slice: ' num2str(slice_Offset_Ref) ', Offset Parameters: [' num2str(imprec_Machine) ' '
num2str(offset_Up_Bound_Machine) '], NUI Kernel Size: ' num2str(nui_Kernel_Size) ', SpanT
Thresh Multiplier: ' num2str(spanT_Thresh_Multiplier)]);

% fprintf('Path drawn. Elapsed time: %g seconds.\n',round(t));

% fprintf(fid,'Path drawn. Elapsed time: %g seconds.\n',round(t));


fig_title = ['Slice ' num2str(slice_Offset_Ref) '_Offset_Params [' num2str(imprec_Machine) ',
num2str(offset_Up_Bound_Machine) ']'_NUI_Kernel_Size ' num2str(nui_Kernel_Size)
'_SpanT_Thresh_X ' num2str(spanT_Thresh_Multiplier)];

imwrite(tiled_UV_Top,['Offset Test\' fig_title '.tif'],'compression','LZW');


% Check whether current slice is OK to use for tiling offsets

dlg_title2 = 'Tiling offset reference slice';

prompt2 = {'Use this slice for tiling offsets (Y/N)?'};

num_lines2 = 1; def2 = {''};

answ2 = inputdlg(prompt2,dlg_title2,num_lines2,def2);

loop_Flag = answ2{1,1};

end


% Save offsets to struct array

spanT_Path_Matrix.(genvarname(['Slice_' num2str(slice_Offset_Ref)])) = spanT_Path;

tile_Offsets_Matrix.(genvarname(['Slice_' num2str(slice_Offset_Ref)])) = tile_Offsets;

global_X_Coords.(genvarname(['Slice_' num2str(slice_Offset_Ref)])) = x_Tile_Coords;

global_Y_Coords.(genvarname(['Slice_' num2str(slice_Offset_Ref)])) = y_Tile_Coords;

```

```

offset_Ref_Params =

struct('slice_Offset_Ref',{slice_Offset_Ref},'nui_Kernel_Size',{nui_Kernel_Size},'nui_Ref_Tile',
{nui_Ref_Tile},'spanT_Thresh_Multiplier',{spanT_Thresh_Multiplier},'imprec_Machine',{imprec_Machine},
'offset_Up_Bound_Machine',{offset_Up_Bound_Machine});

fprintf('\n\tUSING SLICE %i FOR OFFSETS.\n',slice_Offset_Ref);
fprintf(fid,'\n\tUSING SLICE %i FOR OFFSETS.\n',slice_Offset_Ref);

% Parameters for padding and cropping each slice (registration)
% The align_Data here is the max amount of padding needed to align each
% slice, with respect to the slice above it. This is obtained from the
% maximum amount of shift that is required to be performed, to compensate
% for each recentering done on the CNC during imaging. The measurement is
% in pixels.
% Obtain max amount of shift
max_Y_Shift_Pos = 0; max_X_Shift_Pos = 0;
max_Y_Shift_Neg = 0; max_X_Shift_Neg = 0;
Y_Shift_Counter = 0; X_Shift_Counter = 0;
for x = 1 : size(RH_Shift,1)

    X_Shift_Counter = X_Shift_Counter + RH_Shift(x,3);

    % If shifting right, add to max positive x-shift array
    if(RH_Shift(x,3) > 0)

        max_X_Shift_Pos = max_X_Shift_Pos + RH_Shift(x,3);

    % OR if shifting left, add to max negative x-shift array
    elseif (RH_Shift(x,3) < 0)

        max_X_Shift_Neg = max_X_Shift_Neg + RH_Shift(x,3);

```

```

end

Y_Shift_Counter = Y_Shift_Counter + RH_Shift(x,2);

% If shifting down, add to max positive y-shift array
if(RH_Shift(x,2) > 0)

    max_Y_Shift_Pos = max_Y_Shift_Pos + RH_Shift(x,2);

% OR if shifting up, add to max negative y-shift array
elseif (RH_Shift(x,2) < 0)

    max_Y_Shift_Neg = max_Y_Shift_Neg + RH_Shift(x,2);

end

end

% Store maximum shift parameters:

% align_Data should be a 2x2 matrix with y and x maximum negative shifts in
% the first row (respectively) and y and x max positive shifts in the second
align_Data = abs([max_Y_Shift_Neg max_X_Shift_Neg; max_Y_Shift_Pos max_X_Shift_Pos]);

clear tiled_UV_Top

%% 2) Reference Hole Acquisition

fprintf('\n2) Reference Hole Acquisition.\n');

fprintf(fid, '\n2) Reference Hole Acquisition.\n');


fprintf('\n\tNOTE: REFERENCE HOLES NOT USED FOR THIS SPECIMEN.\n');

fprintf(fid, '\n\tNOTE: REFERENCE HOLES NOT USED FOR THIS SPECIMEN.\n');


fprintf('\n\tSlices used for recentering shift:\n');

```

```

fprintf('\tSlice#\tY_Shift(px)\tX_Shift(px)\n');
for x = 1 : size(RH_Shift,1)
    fprintf('\t%i\t%i\t%i\t%i\n',RH_Shift(x,1),RH_Shift(x,2),RH_Shift(x,3));
end

fprintf(fid,'\n\tSlices used for recentering shift:\n');
fprintf(fid,'\tSlice#\tY_Shift(px)\tX_Shift(px)\n');
for x = 1 : size(RH_Shift,1)
    fprintf(fid,'\t%i\t%i\t%i\t%i\n',RH_Shift(x,1),RH_Shift(x,2),RH_Shift(x,3));
end

% Obtain all reference holes
% [ref_Holes,fid] = ...
%
refHoleAcquisition(RH_Shift,y_Tile_Coords,x_Tile_Coords,nui_Ref_Tile,tiles_Edge_Rank_Threshold,
tile_Size,m,n,r,raw_image_dir,code_dir,fid,channel_flag);

% Save struct arrays to disk
save([parameters_out 'spanT_Path_' specimen '_ref_slice-' num2str(slice_Offset_Ref) '.mat'],
'spanT_Path_Matrix');
save([parameters_out 'tile_Offsets_' specimen '_ref_slice-' num2str(slice_Offset_Ref) '.mat'],
'tile_Offsets_Matrix');
save([parameters_out 'global_X_Coords_' specimen '_ref_slice-' num2str(slice_Offset_Ref)
'.mat'], 'global_X_Coords');
save([parameters_out 'global_Y_Coords_' specimen '_ref_slice-' num2str(slice_Offset_Ref)
'.mat'], 'global_Y_Coords');

```

```

save([parameters_out 'align_Data_' specimen '_ref_slice-' num2str(slice_Offset_Ref) '.mat'],
'align_Data');

save([parameters_out 'offset_Ref_Params_' specimen '_ref_slice-' num2str(slice_Offset_Ref)
'.mat'], 'offset_Ref_Params');

%%% 3) Main Processing Algorithm

fprintf('\n3) Main Processing Algorithm.\n');

fprintf(fid, '\n3) Main Processing Algorithm.\n');

% Boolean for finished the first slice

finished_first = false;

% Flag for processing all signals:

% channel_flag=0 for just UV channel

% channel_flag=1 for all three channels

channel_flag = 1;

for i = start_Slice : stop_Slice

    % Main Loop

    % In order to avoid writing this section of code all over again when

    % finished_first = true, we load the bottom only when finished_first =

    % true. In order to avoid skipping a slice each time, load the bottom

    % and rename as top.

    fprintf('\tProcessing slice %i.\n',i);

    fprintf(fid, '\tProcessing slice %i.\n',i);

```

```

tStart = tic;

%% LOAD RAW IMAGES AND CORRECT NUI
fprintf('\t\t-> Loading images & correcting NUI...');
fprintf(fid,'\t\t-> Loading images & correcting NUI...');
tic;

% Top:
if finished_first == false

    [nui_UV_Top,nui_CA_Top,nui_XO_Top] =
loadImagesNUI(top_Num,nui_Ref_Tile,tile_Size,m,n,raw_image_dir,code_dir,channel_flag,nui_
Kernel_Size);

    end

% Bottom

[nui_UV_Bot,nui_CA_Bot,nui_XO_Bot] =
loadImagesNUI(bot_Num,nui_Ref_Tile,tile_Size,m,n,raw_image_dir,code_dir,channel_flag,nui_
Kernel_Size);

t = toc;

fprintf('Finished. Elapsed time: %g seconds.\n',round(t));
fprintf(fid,'Finished. Elapsed time: %g seconds.\n',round(t));

%% TILE AND ALIGN IMAGES
if finished_first == false

    % Tile and align top slice if first time through main loop

    [tiled_UV_Top,tiled_CA_Top,tiled_XO_Top,fid] = ...

```



```

retileAllChannels(nui_UV_Top,nui_CA_Top,nui_XO_Top,y_Tile_Coords,x_Tile_Coords,top_N
um,m,n,fid,channel_flag);

    clear nui_UV_Top nui_CA_Top nui_XO_Top

    [tiled_UV_Top,tiled_CA_Top,tiled_XO_Top,fid] = ...

shiftAllChannels(tiled_UV_Top,tiled_CA_Top,tiled_XO_Top,RH_Shift,align_Data,top_Num,fid
,channel_flag);

end

% Tile and align bottom slice

[tiled_UV_Bot,tiled_CA_Bot,tiled_XO_Bot,fid] = ...

retileAllChannels(nui_UV_Bot,nui_CA_Bot,nui_XO_Bot,y_Tile_Coords,x_Tile_Coords,bot_Nu
m,m,n,fid,channel_flag);

    clear nui_UV_Bot nui_CA_Bot nui_XO_Bot

    [tiled_UV_Bot,tiled_CA_Bot,tiled_XO_Bot,fid] = ...

shiftAllChannels(tiled_UV_Bot,tiled_CA_Bot,tiled_XO_Bot,RH_Shift,align_Data,bot_Num,fid,
channel_flag);

%% OUT OF PLANE SUBTRACTION

fprintf('\t\t-> OP subtraction...');

fprintf(fid,'\t\t-> OP subtraction...');

tic;

tiled_UV_Top = opCorrect(tiled_UV_Top,tiled_UV_Bot,sf_UV,hsize,gspread);

tiled_CA_Top = opCorrect(tiled_CA_Top,tiled_CA_Bot,sf_CA,hsize,gspread);

```

```

    tiled_XO_Top = opCorrect(tiled_XO_Top,tiled_XO_Bot,sf_XO,hsize,gspread);

    t = toc;

    fprintf('Finished. Elapsed time: %g seconds.\n',round(t));

    fprintf(fid, 'Finished. Elapsed time: %g seconds.\n',round(t));


%% THRESHOLD IMAGES

fprintf('\t\t-> Thresholding UV images...');

fprintf(fid, '\t\t-> Thresholding UV images...');

tic;

% Crop out center of tiled mosaic without drill holes to obtain global
% threshold value (drill holes are too bright, and do not produce in a
% good threshold value)

    tiled_UV_Top_Cropped = tiled_UV_Top(:,2*2048:end-2*2048);

%   figure,imshow(tiled_UV_Top_Cropped,[])

% Obtain global threshold value

    global_Thresh_Val =
mean(tiled_UV_Top_Cropped(:))+.7*std(double(tiled_UV_Top_Cropped(:)));

    thresh_Val_Array(i) = global_Thresh_Val;

    fprintf('\n\t\t-> Slice %i: Thresh value = %f...',i,global_Thresh_Val);

    fprintf(fid, '\n\t\t-> Slice %i: Thresh value = %f...',i,global_Thresh_Val);

    thresh_UV_Top = tiled_UV_Top > global_Thresh_Val;

    thresh_UV_Top = bwareaopen(not(thresh_UV_Top),bwa);

    thresh_UV_Top = bwareaopen(not(thresh_UV_Top),bwa);

    final_thresh_UV = imclose(thresh_UV_Top,strel('disk',20));

%   figure,imshow(final_thresh_UV,[]),title('.7');

%   figure,imshow(tiled_UV_Top,[]);

```

```

t = toc;

fprintf('Finished. Elapsed time: %g seconds.\n',round(t));

fprintf(fid,'Finished. Elapsed time: %g seconds.\n',round(t));

%% SAVE PROCESSED IMAGES TO DISK

fprintf('\t\t-> Saving images...');

fprintf(fid,'\t\t-> Saving images...');

tic;

slice_Number = [num2str(zeros(4-length(num2str(i)),1)) num2str(i) '_'];

imwrite(tiled_UV_Top,[tiled_gray_OP_out 'UV_' slice_Number set_rename '_gray.tif'],
'compression', 'LZW');

imwrite(tiled_CA_Top,[tiled_gray_OP_out 'FITC_' slice_Number set_rename '_gray.tif'],
'compression', 'LZW');

imwrite(tiled_XO_Top,[tiled_gray_OP_out 'TRITC_' slice_Number set_rename '_gray.tif'],
'compression', 'LZW');

imwrite(thresh_UV_Top,[tiled_thresh_out 'UV_' slice_Number set_rename
'_thresh.tif'],'compression','LZW');

t = toc;

fprintf('Finished. Elapsed time: %g seconds.\n',round(t));

fprintf(fid,'Finished. Elapsed time: %g seconds.\n',round(t));

% Increment offset and move bottom images to top

bot_Num = bot_Num + 1;

clear tiled_UV_Top tiled_CA_Top tiled_XO_Top thresh_UV_Top

tiled_UV_Top = tiled_UV_Bot; tiled_CA_Top = tiled_CA_Bot; tiled_XO_Top =
tiled_XO_Bot;

```

```

clear tiled_UV_Bot tiled_CA_Bot tiled_XO_Bot

tEnd = toc(tStart);

fprintf('\t--> Slice finished: %i. Elapsed time: %g seconds.\n\n',i,round(tEnd));

fprintf(fid, '\t--> Slice finished: %i. Elapsed time: %g seconds.\n\n',i,round(tEnd));

%% Finished first slice

finished_first = true;

%% Save parameters

save([parameters_out 'thresh_Val_Array_' specimen '_' num2str(start_Slice) '-'
num2str(stop_Slice) '.mat'], 'thresh_Val_Array');

end

fprintf(fid, '\n\nFinished specimen %s: slices %i-%i,
%s\n\n',specimen,start_Slice,stop_Slice,datestr(clock));

fclose(fid);

cd(code_dir)

```

GetSpanTree.m

```

%% getSpanT.m
% This function calculates the minimum spanning tree path for a given set
% of images.

%% ~~~~~INPUT~~~~~
% nui_Images -

```

```

%% ~~~~~OUTPUT~~~~~
% tiling_Order -

% EVT 04.22.11
% EVT 04.27.11

function [edge_Matrix,edge_Tiling_Order,spanT_Path] =
getSpanT(nui_Images,m,n,tile_size,offset_Up_Bound_Machine,edge_Rank_Thresh_Val,bwa)

% Create edge matrix - index of matrix corresponds to the edge between
% the tiles in columns 1 and 2; the edges are ordered from left to right,
% and then from top to bottom (i.e. edge #1 is between tiles 1 and 2,
% edge #2 is between tiles 1 and 7); EXAMPLE: an index of 11,
% edge_Matrix(11,:), corresponds to the 11th edge, between tiles 6 and
% 12; the third column of edge_Matrix will be populated as the rank
% (order calculated by the amount of black pixels in each edge)
edge_Matrix = createEdgeMatrix(m,n);

% Calculate amount of black space on the tile edges
pixel_Vector = zeros(length(edge_Matrix),1);

% Indexes in pixel_Vector
counter = 0;

for i = 1:m*n

    % Obtain first tile for examination
    iTile1 = nui_Images(:,i);
    iTile1 = iTile1 > edge_Rank_Thresh_Val;
    iTile1 = bwareaopen(not(iTile1),100);
    iTile1 = bwareaopen(not(iTile1),100);

    % If the tile is not in the right column, crop right edge
    if mod(i, 6) ~= 0
        % Obtain next adjacent tile to the right side
        iTile2 = nui_Images(:,i+1);
        iTile2 = iTile2 > edge_Rank_Thresh_Val;
        iTile2 = bwareaopen(not(iTile2),100);
        iTile2 = bwareaopen(not(iTile2),100);

        % Crop appropriate tiles
        iTile_right_edge = iTile1(:,tile_size(2)-offset_Up_Bound_Machine+1 : tile_size(2));
        iTile_left_edge = iTile2(:,1:round(offset_Up_Bound_Machine/2));
        counter = counter+1;
        pixel_Vector(counter) = calcMaterial(iTile_right_edge,iTile_left_edge);

        clear iTile_right_edge iTile_left_edge iTile2

    % If the tile is not in the bottom row, crop bottom edge
    if i <= (m-1)*n
        % Obtain next adjacent tile underneath, at the bottom side

```

```

iTile2 = nui_Images(:,:,i+m);
iTile2 = iTile2 > edge_Rank_Thresh_Val;
iTile2 = bwareaopen(not(iTile2),100);
iTile2 = bwareaopen(not(iTile2),100);

iTile_bottom_edge = iTile1(tile_size(1)-offset_Up_Bound_Machine+1 : tile_size(1),:);
iTile_top_edge = iTile2(1:round(offset_Up_Bound_Machine/2),:);
counter = counter+1;
pixel_Vector(counter) = calcMaterial(iTile_bottom_edge,iTile_top_edge);

clear iTile_bottom_edge iTile_top_edge iTile2
end
% If the tile is in the right column, do not crop right edge
else
% If the tile is not in the bottom row, crop bottom edge
if i <= (m-1)*n
% Obtain next adjacent tile underneath, at the bottom side
iTile2 = nui_Images(:,:,i+m);
iTile2 = iTile2 > edge_Rank_Thresh_Val;
iTile2 = bwareaopen(not(iTile2),100);
iTile2 = bwareaopen(not(iTile2),100);

iTile_bottom_edge = iTile1(tile_size(1)-offset_Up_Bound_Machine+1 : tile_size(1),:);
iTile_top_edge = iTile2(1:round(offset_Up_Bound_Machine/2),:);
counter = counter+1;
pixel_Vector(counter) = calcMaterial(iTile_bottom_edge,iTile_top_edge);

clear iTile_bottom_edge iTile_top_edge iTile2
end
end
clear iTile1
end

% Convert pixel_Number into edge ranks (this is a sorted list in
% descending order of black pixels in right or bottom edge of each tile)
edge_Rank = pixel2rank(pixel_Vector);

% Insert rank (sorted by amount of black space) into column 3 of edge_Matrix
edge_Matrix(:,3) = edge_Rank;

% Solve spanning tree algorithm to obtain minimum spanning tree
edge_Tiling_Order = grMinSpanTree(edge_Matrix);
spanT_Path = edge_Matrix(edge_Tiling_Order,:);

end

```

Main.m (Rat vertebrae)

```

Orange - XOT','Area Opening Size - BWA'};
dlg_title = 'LPPC Settings (default values)';
num_lines = [1];
def = {'0.2','50','15','25','20','.95','.6','.6','100'};
answ = inputdlg(prompt,dlg_title,num_lines,def);
sf = str2double(answ{1,1});
hsize = str2double(answ{2,1});
gspread = str2double(answ{3,1});
gsize = str2double(answ{4,1});
its = str2double(answ{5,1});
UVT = str2double(answ{6,1});
CAT = str2double(answ{7,1});
XOT = str2double(answ{8,1});
bwa = str2double(answ{9,1});
disp('Press enter to begin processing. ');
q = input("");
disp(['Processing ' setname]);
%% 3) Fiduciary Marker Offset Determination
tic;
i = (m*n*(globalstart-1)+1);
for j = 1:m*n
    images(:,j) = imread(files(i+j-1).name);
end
images = NUI_Correct(images,m,n);
[tildeUV,offset,ccvalue] = Tiling(images, m, n, pixelsize);
[refholes threshvalue] = Locate_Fiducial(tiledUV, type, r, m, n, val);
cropdata = (1/val)*Crop(refholes,aligndata,type,m,n,(size(images)*val));
disp('Reference Offset Acquired. '); t = toc;
disp( [' Elapsed time: ' num2str(round(t)) ' seconds']);
%% To restart, uncomment this cell and run to the end (F9)
% cd(threshdir);
% sets = dir('*.*mat');
% for i=1:length(sets)
% load(sets(i).name);
% end
% cd(imagedir);
% h=start;
% finishedfirst = false;
% clc
%% 4) Main Loop
tic;
for i=(m*n*(start-1)+1) : m*n : (m*n*(stop-1)+1)
    files = dir( '*UV*' );
    %First loop index
    if finishedfirst==false
        finishedfirst = true;
    %Top
    loaded = false;
    while loaded==false
        try
            for j = 1:m*n

```

```

imagesUV(:,j) = imread(files(i+(j-1)).name);
imagesCA(:,j) = imread([files(i+(j-1)).name(1:8) '_FITC_' setname]);
imagesXO(:,j) = imread([files(i+(j-1)).name(1:8) '_TRITC_' setname]);
end
loaded=true;
catch
disp(['Files not available after ' files(i).name(1:5)]);
disp('Resuming processing in 5 minutes');
pause(300);
end
end
%Bottom
loaded = false;
while loaded==false
try
for j = 1:m*n
imagesUV2(:,j) = imread(files(i+(j-1)+m*n).name);
imagesCA2(:,j) = imread([files(i+(j-1)+m*n).name(1:8) '_FITC_' setname]);
imagesXO2(:,j) = imread([files(i+(j-1)+m*n).name(1:8) '_TRITC_' setname]);
end
loaded=true;
catch
disp(['Files not available after ' files(i).name(1:5)]);
disp('Resuming processing in 5 minutes');
pause(300);
end
end
%Non uniform illumination correction
imagesUV = NUI_Correct(imagesUV,m,n);
imagesCA = NUI_Correct(imagesCA,m,n);
imagesXO = NUI_Correct(imagesXO,m,n);
imagesUV2 = NUI_Correct(imagesUV2,m,n);
imagesCA2 = NUI_Correct(imagesCA2,m,n);
imagesXO2 = NUI_Correct(imagesXO2,m,n);
%Mosaic tiling
%UV Top Image only, find alignment and tiling offsets
[tiledUV,tileoffset(:,h),ccvalues] = Tiling(imagesUV, m, n, pixelsize);
offmarker = Fiducial_Selection_2(type, ccvalues, m, n);
alignoffset(h, :) =
(1/val)*Fiducial_Selection_3(tiledUV,refholes(offmarker).subs,type(offmarker,:),threshvalue,length(type),m,n,r,val);
disp( [ ' Alignment Offset: Slice ' files(i).name(1:5) ' [ ' num2str(alignoffset(h,:)) ' ] ] );
%Top Signals - tile, pad, align, and crop
tiled_top = Retile(imagesUV,imagesCA,imagesXO,tileoffset(:,h),m,n);
tiled_top = Pad_Crop(tiled_top,alignoffset(h,:),aligndata,cropdata);
%UV Bottom Image only, find alignment and tiling offsets
[tiledUV,tileoffset(:,h+1),ccvalues] = Tiling(imagesUV2, m, n, pixelsize);
offmarker = Fiducial_Selection_2(type, ccvalues, m, n);
alignoffset(h+1, :) =
(1/val)*Fiducial_Selection_3(tiledUV,refholes(offmarker).subs,type(offmarker,:),threshvalue,length(type),m,n,r,val);

```



```

disp( [ ' Alignment Offset: Slice ' files(i+m*n).name(1:5) ' [ ' num2str(alignoffset(h+1,:)) ' ] ] );
% Correct alignment offsets
if abs((alignoffset(h+1,1)+1000) - (alignoffset(h,1)+1000)) > maxalign
alignoffset(h+1,1) = alignoffset(h,1);
disp(['Y offset corrected: [ ' num2str(alignoffset(h,1)) ' ]]);
end
if abs((alignoffset(h+1,2)+1000) - (alignoffset(h,2)+1000)) > maxalign
alignoffset(h+1,2) = alignoffset(h,2);
disp(['X offset corrected: [ ' num2str(alignoffset(h,2)) ' ]]);
end
% Bottom Signals - tile, pad, align, and crop
tiled_bot = Retile(imagesUV2,imagesCA2,imagesXO2,tileoffset(:,h+1),m,n);
tiled_bot = Pad_Crop(tiled_bot,alignoffset(h+1,:),aligndata,cropdata);
%Subsurface fluorescence removal
tiled_top = SubSurf_Fluor_Correct(tiled_top,tiled_bot,sf,hsize,gspread);
%Segmentation
UVthresh = Threshold(tiled_top(:,1),lppc_itp(tiled_top(:,1))*UVT,gsize,its);
CAthresh = ( tiled_top(:, 2) > max(max(tiled_top(:, 2)))*CAT );
XOthresh = ( tiled_top(:, 3) > max(max(tiled_top(:, 3)))*XOT );
%Post Processing on Images to remove marrow space noise
UVthresh = bwareaopen(not(UVthresh),bwa);
UVthresh = bwareaopen(not(UVthresh),bwa);
CAthresh = bwareaopen(not(CAthresh),bwa);
CAthresh = bwareaopen(not(CAthresh),bwa);
XOthresh = bwareaopen(not(XOthresh),bwa);
XOthresh = bwareaopen(not(XOthresh),bwa);
% Save images to disk
imwrite( tiled_top(:,1), [ threshdir 'UV\ ' files(i).name(1:5) 'UV_ ' setname '.tif', 'Compression',
'none' ] );
imwrite( UVthresh, [threshdir files(i).name(1:5) 'UV_ ' setname '_thresh.tif', 'Compression',
'none' ] );
imwrite( CAthresh, [threshdir files(i).name(1:5) 'FITC_ ' setname '_thresh.tif', 'Compression',
'none' ] );
imwrite( XOthresh, [threshdir files(i).name(1:5) 'TRITC_ ' setname '_thresh.tif', 'Compression',
'none' ] );
% Increment offset and move bottom images to top
h = h + 1;
tiled_top = tiled_bot;
t=toc;
timeleft = (round(10*(((stop-str2num(files(i).name(1:5)))*(t/(str2num(files(i).name(1:5))-
start)))/60)/60)/10);
disp(['---> Finished Slice ' files(i).name(1:5) ' . ' num2str(timeleft) ' hours remaining.']);
else
%All loop except first
tic;
%Bottom
if (round(i/(m*n))+1) ~= stop %only try to load bot files if its not at the end of the run
loaded = false;
while loaded==false
try
for j = 1:m*n

```

```

imagesUV2(:,j) = imread(files(i+(j-1)+m*n).name);
imagesCA2(:,j) = imread([files(i+(j-1)+m*n).name(1:8) '_FITC_' setname]);
imagesXO2(:,j) = imread([files(i+(j-1)+m*n).name(1:8) '_TRITC_' setname]);
end
loaded=true;
catch
disp(['Files not available after ' files(i).name(1:5)]);
disp('Resuming processing in 5 minutes');
pause(300);
end
end
%Non uniform illumination correction
imagesUV2 = NUI_Correct(imagesUV2,m,n);
imagesCA2 = NUI_Correct(imagesCA2,m,n);
imagesXO2 = NUI_Correct(imagesXO2,m,n);
%Tiling
% UV Bottom Image only, find alignment and tiling offsets
[tiledUV,tileoffset(:,h+1),ccvalues] = Tiling(imagesUV2, m, n, pixelsize);
offmarker = Fiducial_Selection_2(type, ccvalues, m, n);
alignoffset(h+1, :) =
(1/val)*Fiducial_Selection_3(tiledUV,refholes(offmarker).subs,type(offmarker,:),threshvalue,len
gth(type),m,n,r,val);
disp( [ ' Alignment Offset: Slice ' files(i+m*n).name(1:5) ' [' num2str(alignoffset(h+1,:)) ']' ] );
% Correct alignment offsets
if abs((alignoffset(h+1,1)+1000) - (alignoffset(h,1)+1000)) > maxalign
alignoffset(h+1,1) = alignoffset(h,1);
disp(['Y offset corrected: [' num2str(alignoffset(h,1)) ']]);
end
if abs((alignoffset(h+1,2)+1000) - (alignoffset(h,2)+1000)) > maxalign
alignoffset(h+1,2) = alignoffset(h,2);
disp(['X offset corrected: [' num2str(alignoffset(h,2)) ']]);
end
% Bottom Signals - tile, pad, align, and crop
tiled_bot = Retile(imagesUV2,imagesCA2,imagesXO2,tileoffset(:,h+1),m,n);
tiled_bot = Pad_Crop(tiled_bot,alignoffset(h+1,:),aligndata,cropdata);
%Subsurface fluorescence correction
tiled_top = SubSurf_Fluor_Correct(tiled_top,tiled_bot,sf,hsize,gspread);
end
% Segmentation
UVthresh = Threshold(tiled_top(:,1),lppc_itp(tiled_top(:,1))*UVT,gsize,its);
CAthresh = ( tiled_top(:, 2) > max(max(tiled_top(:, 2))))*CAT );
XOthresh = ( tiled_top(:, 3) > max(max(tiled_top(:, 3))))*XOT );
%Post Processing on Images to remove marrow space noise
UVthresh = bwareaopen(not(UVthresh),bwa);
UVthresh = bwareaopen(not(UVthresh),bwa);
CAthresh = bwareaopen(not(CAthresh),bwa);
CAthresh = bwareaopen(not(CAthresh),bwa);
XOthresh = bwareaopen(not(XOthresh),bwa);
XOthresh = bwareaopen(not(XOthresh),bwa);
% Save images to disk

```

```

imwrite( tiled_top(:,1), [ threshdir 'UV\' files(i).name(1:5) 'UV_' setname '.tif'], 'Compression',
'none' );
imwrite( UVthresh, [threshdir files(i).name(1:5) 'UV_' setname '_thresh.tif'], 'Compression',
'none' );
imwrite( CAthresh, [threshdir files(i).name(1:5) 'FITC_' setname '_thresh.tif'], 'Compression',
'none' );
imwrite( XOthresh, [threshdir files(i).name(1:5) 'TRITC_' setname '_thresh.tif'], 'Compression',
'none' );
save( [threshdir 'refholes_' num2str(start) '-' num2str(stop) '.mat' ], 'refholes' );
save( [threshdir 'tileoffset_' num2str(start) '-' num2str(stop) '.mat' ], 'tileoffset' );
save( [threshdir 'alignoffset_' num2str(start) '-' num2str(stop) '.mat' ], 'alignoffset' );
% Increment offset and move bottom images to top
h = h + 1;
tiled_top = tiled_bot;
t=toc;
timeleft = (round(10*(((stop-str2num(files(i).name(1:5)))*(t/(str2num(files(i).name(1:5))-
start)))/60)/60))/10);
disp(['---> Finished Slice ' files(i).name(1:5) ' ' num2str(timeleft) ' hours remaining.']);
end
end
%%% Post Processing
disp(['Beginning Post Processing on ' setname]);
se = zeros(3,3,3); se(2,2,:) = 1;
cd(threshdir);
files = dir('*UV_*');
for i=1:length(files)
test(:,1) = imread(files(i).name);
test(:,2) = imread(files(i+1).name);
test(:,3) = imread(files(i+2).name);
output = imopen(test,se);
imwrite( output(:,1), [ threshdir files(i).name], 'Compression', 'none' );
disp(['Finished with' num2str(i)]);
end

```

NUI_Correct1.m

```

% Non uniform illumination correction
% COPYRIGHT August 2008
% Craig R Slyfield Jr., Ryan E. Tomlinson
% Corrects for non uniform illumination by division of an illumination profile
% fit. Requires Illumination_Profiler.m
% ADDITION: Verify the edge intensities and make corrections as necessary.
function img = NUI_Correct1(images, m, n)
%%% Allocate variables and constants
H = size(images,1); W = size(images,2); D = size(images,3); overlap = 30;
timage = zeros(H,W,'double'); img = zeros(H,W,D,'uint16'); coeff = zeros(6,D);
B = 1.2; C = 3; E = 5000;
%%% Find Coefficients of background illumination profile
for q=1:D
coeff(:,q) = Illumination_Profiler(images(:, :, q));

```

```

end
a = (sum(coeff,2)' ./ D);
a(1) = a(1)*B;
a(2) = a(2)*B;
a(3) = a(3)*B;
%%Divide image by illumination profile
for o=1:D
N = images(:, :, o);
x=1:W;
for y=1:H
timage(y,x) = double(N(y,x)) ./ (a(1) + a(2).*y + a(3).*x + a(4).*y.*y + a(5).*x.*x + a(6).*y.*x);
end
img(:, :, o) = (timage+C)*E;
end
%% Verify edge intensities
%Horizontal
for j=1:m
for i=1:(n-1)
index = i+(j-1)*n;
diff = mean(mean( img(3*overlap:end-3*overlap, W-overlap : W-overlap+5, index) )) -
mean(mean( img(3*overlap:end-3*overlap, overlap+1-5 : overlap+1, index+1) ));
img(:, :, index+1) = img(:, :, index+1) + 0.7*diff;
end
end
%Vertical
for j=1:(m-1)
for i=1:n
index = i+(j-1)*n;
diff = mean(mean( img(end-7:end, :, index) )) - mean(mean( img(1:8, :, index+n) ));
img(:, :, index+n) = img(:, :, index+n) + 0.7*diff;
end
end
end

```

Tiling.m

```

function [ tiledimg, offsets, ccvalues ] = Tiling( images, m, n, pixelsize)
% COPYRIGHT August 2008,
% Craig R Slyfield Jr., Ryan E. Tomlinson, Kyle E. Niemeyer
%=====
% This function assembles an arbitrarily sized image array
%
% Include Determine_Offset.m , Tile.m
%
%=====
%% Tiling of images in rows
y = size( images(:, :, 1), 1 ); x = size( images(:, :, 1), 2 );
if strcmp( class( images ), 'double' )
rowtiled = zeros( y + 10, x*n, m );
elseif strcmp( class( images ), 'uint8' )
rowtiled = uint8( zeros( y + 10, x*n, m ) );

```

```

elseif strcmp( class( images ), 'uint16' )
rowtiled = uint16( zeros( y + 10, x*n, m ) );
end
% The offsets in the y direction should sum to no more than 10, and the x
% offsets are always going to make the tiled image smaller than the sum of
% the widths
if (n - 1) > (m - 1)
ccvalues = zeros( (m + 1), (n - 1) );
offsets = zeros( m + 1, (n - 1)*2 );
else
ccvalues = zeros( (m + 1), (m - 1) );
offsets = zeros( m + 1, (m - 1)*2 );
end
% Gets horizontal offsets
for l = 1 : m
img1 = images( :, :, (l-1)*n + 1 );
t = 1; % Index for offset matrix
c = 1; % Index for cc value matrix
for j = 2 : n
img2 = images( :, :, (l-1)*n + j );
% Adjust canvas size to accomodate registration (only necessary in
% some cases)
bigsize = size(img1);
if size( img2, 1 ) > size( img1, 1 );
bigsize( 1, 1 ) = size( img2, 1 );
end
if size( img2, 2 ) > size( img1, 2 );
bigsize( 1, 2 ) = size( img2, 2 );
end
if strcmp( class( images ), 'double' )
img1dum = zeros(bigsize); img2dum = zeros(bigsize);
elseif strcmp( class( images ), 'uint16' )
img1dum = uint16(zeros(bigsize)); img2dum = uint16(zeros(bigsize));
elseif strcmp( class( images ), 'uint8' )
img1dum = uint8(zeros(bigsize)); img2dum = uint8(zeros(bigsize));
end
img1dum( 1 : size( img1, 1 ), 1 : size( img1, 2 ) ) = img1;
img2dum( 1 : size( img2, 1 ), 1 : size( img2, 2 ) ) = img2;
img1 = img1dum; img2 = img2dum;
[ offset, cc ] = Determine_Offset( img1, img2, pixelsize );
if offset == [3 18] % Dummy offset from register.m
if ( (l-1)*n + j ) > (m-1)*n
[ offset, cc ] = Determine_Offset( images( :, :, (l-2)*n + j - 1 ), images( :, :, (l-2)*n + j ), pixelsize);
else
[ offset, cc ] = Determine_Offset( images( :, :, l*n + j - 1 ), images( :, :, l*n + j ), pixelsize);
end
end
end
% If dummy offset is given, finds offset at different interface (where
% there is hopefully more bone material
row = Tile( img1, img2, offset );
%added this to get rid of the blank menace - RET 2/1/08

```

```

temprow = row(15,:);
[I J] = find(temprow<1);
Try
A = find(J>size(row,2)*.65);
newJ = J(A);
if newJ(1) > size(row,2)/2
row(:,newJ(1):newJ(end)) = [];
end
catch
end
offsets( 1, t : t + 1 ) = offset;
t = t + 2;
ccvalues( 1, c ) = cc;
c = c + 1;
img1 = row;
end
rowtiled( 1 : size( row, 1 ), 1 : size( row, 2 ), 1 ) = row;
end
%% Tiling of rows together
row1 = rowtiled( :, :, 1 );
p = 0;
for k = size(row1,1) : -1 : size(row1,1) - 100
if max( row1( k, : ) ) == 0
p = k;
end
end
if p ~= 0
row1( p : end, : ) = [];
end
p = 0;
for k = 1 : 50
if max( row1( k, : ) ) == 0
p = k;
end
end
if p ~= 0
row1( 1 : p, : ) = [];
end
% Assignment and cleanup of first row
if m == 1
tiledimg = row1;
end
t = 1; c = 1; % Indices for offsets and cc matrices
for l = 2 : m
row2 = rowtiled( :, :, l );
p = 0;
for k = size(row2, 1) : -1 : size(row2, 1) - 100
if max( row2( k, : ) ) == 0
p = k;
end
end
end
end

```

```

if p ~= 0
row2( p : end, : ) = [];
end
p = 0;
for k = 1 : 50
if max( row2( k, : ) ) == 0
p = k;
end
end
if p ~= 0
row2( 1 : p, : ) = [];
end
% Adjust canvas size to accomodate registration (only necessary in
% some cases)
bigsize = size(row1);
if size( row2, 1 ) > size( row1, 1 )
bigsize( 1, 1 ) = size( row2, 1 );
end
if size( row2, 2 ) > size( row1, 1 )
bigsize( 1, 2 ) = size( row2, 2 );
end
if strcmp( class( row1 ), 'double' )
row1dum = zeros(bigsize); row2dum = zeros(bigsize);
elseif strcmp( class( row1 ), 'uint16' )
row1dum = uint16(zeros(bigsize)); row2dum = uint16(zeros(bigsize));
elseif strcmp( class( row1 ), 'uint8' )
row1dum = uint8(zeros(bigsize)); row2dum = uint8(zeros(bigsize));
end
row1dum( 1 : size( row1, 1 ), 1 : size( row1, 2 ) ) = row1;
row2dum( 1 : size( row2, 1 ), 1 : size( row2, 2 ) ) = row2;
row1 = row1dum; row2 = row2dum;
[ offset, cc ] = Determine_Offset( row1', row2', pixelsize);
p = 1;
while offset == [3 18] % dummy offset
img1 = images( :, :, (l-2)*n + p );
img2 = images( :, :, (l-1)*n + p );
[ offset, cc ] = Determine_Offset( img1', img2', pixelsize );
p = p + 1;
end
tiledimg = ( Tile( row1', row2', offset ) )';
offsets( m + 1, t : t + 1 ) = offset;
t = t + 2;
ccvalues( m + 1, c ) = cc;
c = c + 1;
if l > 2
pos = size( tiledimg, 1 ) - ( l - 2 ) * y;
ind = 1;
while ind ~= 0
ind = max( tiledimg( pos, : ) );
pos = pos + 1;
end
end

```

```

tiledimg( pos : end, : ) = [];
end
% Cleanup of black space necessary for registration
row1 = tiledimg;
end
%% Final cleanup of tiled images
x = size( tiledimg, 2 ); y = size( tiledimg, 1 );
p = 0;
for k = x : -1 : x - ( x/3 )
if max( tiledimg( :, k ) ) == 0
p = k;
end
end
if p ~= 0
tiledimg( :, p : end ) = [];
end
p = 0;
for k = y : -1 : y - ( y/3 )
if max( tiledimg( k, : ) ) == 0
p = k;
end
end
if p ~= 0
tiledimg( p : end, : ) = [];
end
end

```

Fiducial_Selection.m

```

%COPYRIGHT August 2008
% Craig R. Slyfield Jr., Ryan E. Tomlinson, Kyle E. Niemeyer
function type = Fiducial_Selection(m,n,img)
%=====
% Fiduciary marker position selection tool
%
%=====
fidmarkerdum = imresize(img, [m*100 n*100]);
block = 65.5e3*ones(20,20);
fidmarkerdum( 1 : 20, 1 : 20 ) = block;
fidmarkerdum( size(fidmarkerdum, 1) - 19 : size(fidmarkerdum, 1), 1 : 20 ) = block;
fidmarkerdum( size(fidmarkerdum, 1) - 19 : size(fidmarkerdum, 1), size(fidmarkerdum, 2) - 19 :
size(fidmarkerdum, 2) ) = block;
fidmarkerdum( 1 : 20, size(fidmarkerdum, 2) - 19 : size(fidmarkerdum, 2) ) = block;
for nn = 1 : n - 147
fidmarkerdum( 1 : 20, nn*100 - 9 : nn*100 + 10 ) = block;
fidmarkerdum( :, nn*100 ) = 0;
fidmarkerdum( size(fidmarkerdum, 1) - 19 : size(fidmarkerdum, 1), nn*100 - 9 : nn*100 + 10 ) =
block;
end
for mm = 1 : m - 1
fidmarkerdum( mm*100 - 9 : mm*100 + 10, 1 : 20 ) = block;

```



```

fidmarkerdum( mm*100, : ) = 0;
fidmarkerdum( mm*100 - 9 : mm*100 + 10, size(fidmarkerdum, 2) - 19 : size(fidmarkerdum, 2)
) = block;
end
figure(1); imshow(fidmarkerdum,[]);
prompt = {'Number of fiduciary markers to use:'};
dlg_title = 'User Input';
num_lines = 1;
def = {''};
answ = inputdlg(prompt,dlg_title,num_lines,def);
num = str2num(answ{1,1});
figure(1); imshow( fidmarkerdum, []), title( 'Click in the positions of the drill holes:' )
positions = ginput( num ); close
% This will isolate the fiduciary marker if the positions are located
% somewhere within each marker
type = zeros( num, 2 );
for kk = 1 : num
pos = positions( kk, : );
for nn = 0 : n
if ( pos(1) > nn*100 - 19 ) && ( pos(1) < nn*100 + 20 )
type( kk, 2 ) = nn;
end
end
for mm = 0 : m
if ( pos(2) > mm*100 - 19 ) && ( pos(2) < mm*100 + 20 )
type( kk, 1 ) = mm;
end
end
end
% This labels the fiducials with the positions based on interface,
% ranging from (0,0) to (m,n)

```

NUI_Correct.m

```

%COPYRIGHT August 2008,
% Craig R Slyfield Jr., Ryan E. Tomlinson
% Corrects for non uniform illumination over an image by division of a background illumination
profile fit
% Include Illumination_Profiler.m
function img = NUI_Correct(images, m, n)
%% Allocate variables and constants
H = size(images,1); W = size(images,2); D = size(images,3); overlap = 30;
timage = zeros(H,W,'double'); img = zeros(H,W,D,'uint16'); coeff = zeros(6,D);
B = 1; C = 3; E = 5000;
%% Find background illumination profile coefficients
for q=1:D
coeff(:,q) = Illumination_Profiler(images(:, :, q));
end
a = (sum(coeff,2)' ./ D);
a(1) = a(1)*B;

```

```

a(2) = a(2)*B;
a(3) = a(3)*B;
%% Divide original image by the fitted background illumination profile fit
for o=1:D
N = images(:, :, o);
x=1:W;
for y=1:H
timage(y,x) = double(N(y,x)) ./ (a(1) + a(2).*y + a(3).*x + a(4).*y.*y + a(5).*x.*x + a(6).*y.*x);
end
img(:, :, o) = (timage+C)*E;
end
%% Verify edge intensities
%Horizontal
for j=1:m
for i=1:(n-1)
index = i+(j-1)*n;
diff = mean(mean( img(3*overlap:end-3*overlap, W-overlap : W-overlap+5, index) )) -
mean(mean( img(3*overlap:end-3*overlap, overlap+1-5 : overlap+1, index+1) ));
img(:, :, index+1) = img(:, :, index+1) + 0.5*diff;
end
end
%Vertical
for j=1:(m-1)
for i=1:n
index = i+(j-1)*n;
diff = mean(mean( img(end-7:end, :, index) )) - mean(mean( img(1:8, :, index+n) ));
img(:, :, index+n) = img(:, :, index+n) + 0.5*diff;
end
end

```

Locate_Fiducial.m

```

%COPYRIGHT August 2008,
% Craig R Slyfield Jr., Ryan E. Tomlinson
function [reference_Holes thresh_Value] = Locate_Fiducial( img, type, r, m, n, val )
% =====
% This function allows the user to manually select the approximate positions of fiduciary markers
% =====
resized_Img = imresize(img, val );
yy = size(resized_Img, 1);
xx = size(resized_Img, 2);
ys = round( size(resized_Img, 1) / m );
xs = round( size(resized_Img, 2) / n );
rs = round( r * val );
ref_Circle = ones(2*rs, 2*rs);
for yi = 1 : 2*rs
for xi = 1 : 2*rs
if ( sqrt( ( yi - rs )^2 + ( xi - rs )^2 ) ) > (rs - .5)
ref_Circle( yi, xi ) = 0;
end
end
end
end

```

```

reference_Holes( length(type) ) = struct('img', {0}, 'subs', {0}, 'norm_Offsets', {0} );
thresh_Holes = single( zeros( length(type), 1 ) );
for hole = 1 : size(type,1)
if type(hole, 1) == 0 %Along top
yrange = [1 20];
if type(hole, 2) == 0 %Upper left
xrange = [1 20];
elseif type(hole, 2) == n %Upper right
xrange = [(xx - 21) xx];
else
xrange = [(xs * type(hole, 2) - 10) (xs * type(hole, 2) + 10)];
end
elseif type(hole, 1) == m
%Along bottom
yrange = [(yy - 21) yy];
if type(hole, 2) == 0 %Bottom left
xrange = [1 20];
elseif type(hole, 2) == n %Bottom right
xrange = [(xx - 21) xx];
else
xrange = [(xs * type(hole, 2) - 10) (xs * type(hole, 2) + 10)];
end
elseif type(hole, 2) == 0 %Along left
xrange = [1 20];
yrange = [(ys * type(hole, 1) - 10) (ys * type(hole, 1) + 10)];
elseif type(hole, 2) == n %Along right
xrange = [(xx - 21) xx];
yrange = [(ys * type(hole, 1) - 10) (ys * type(hole, 1) + 10)];
else %Anywhere else
xrange = [(xs * type(hole, 2) - 10) (xs * type(hole, 2) + 10)];
yrange = [(ys * type(hole, 1) - 10) (ys * type(hole, 1) + 10)];
end
thresh_Holes(hole) = mean(mean( resized_Img( yrange(1) : yrange(2), xrange(1) : xrange(2) ) ));
end
%-----%
k=0.6;
%-----%
thresh_Value = k*max(resized_Img(:));
drill_Holes_img = resized_Img > thresh_Value;
drill_Holes_img = bwareaopen(not(drill_Holes_img), 5000);
drill_Holes_img = bwareaopen(not(drill_Holes_img), 5000);
y_Positions = type(:, 1)*ys + 1;
x_Positions = type(:, 2)*xs + 1;
for i = 1 : size(type,1)
if y_Positions( i ) > yy
y_Positions( i ) = yy;
end
if x_Positions( i ) > xx
x_Positions( i ) = xx;
end
end

```

```

drill_Holes_temp =
imfill(drill_Holes_img,sub2ind(size(drill_Holes_img),y_Positions(i),x_Positions(i)));
% In case the fill operation fills the entire image, check the middle
if drill_Holes_temp(round((size(drill_Holes_temp,1)/2)),round((size(drill_Holes_temp,2)/2))) ~=
1
drill_Holes_img = drill_Holes_temp;
end
drill_Holes_img = bwareaopen(not(drill_Holes_img), 500);
drill_Holes_img = bwareaopen(not(drill_Holes_img), 500);
drill_Holes_img = imopen(drill_Holes_img,strel('disk',10));
drill_Holes_img = imclose(drill_Holes_img,strel('disk',10));
drill_Holes_img = bwareaopen(not(drill_Holes_img),500);
drill_Holes_img = bwareaopen(not(drill_Holes_img),500);
[reference_Holes(i).img reference_Holes_indices] = bwselect( drill_Holes_img, x_Positions(i),
y_Positions(i) );
reference_Holes_subs = zeros( length( reference_Holes_indices ), 2 );
[reference_Holes_subs(:, 1) reference_Holes_subs(:, 2)] = ind2sub( [yy xx],
reference_Holes_indices );
reference_Holes(i).subs = zeros(2, 2);
reference_Holes(i).subs(1, 1) = min( reference_Holes_subs(:, 1) );
reference_Holes(i).subs(2, 1) = max( reference_Holes_subs(:, 1) );
reference_Holes(i).subs(1, 2) = min( reference_Holes_subs(:, 2) );
reference_Holes(i).subs(2, 2) = max( reference_Holes_subs(:, 2) );
reference_Holes(i).img = reference_Holes(i).img( reference_Holes(i).subs(1, 1) :
reference_Holes(i).subs(2, 1), reference_Holes(i).subs(1, 2) : reference_Holes(i).subs(2, 2) );
% %
off = fidmarkreg( ref_Circle, reference_Holes(i).img );
% %
if reference_Holes(i).subs(1,1) < 1
reference_Holes(i).subs(1,1) = 1;
end
if reference_Holes(i).subs(1,2) < 1
reference_Holes(i).subs(1,2) = 1;
end
reference_Holes(i).norm_Offsets = off;
end

```

Crop.m

```

%COPYRIGHT August 2008,
% Craig R. Slyfield Jr., Ryan E. Tomlinson, Kyle E. Niemeyer
function cropdata = Crop( refholes, aligndata, type, m, n, imgsize)
% =====
% INPUT: Fiducial Information (reference_Holes)
% a structure with .img, .subs, and .norm_Offsets
% Alignment Data (align_Data)
% A 2x2 matrix of the absolute value of max and min offset
% from fiduciary marker alignment
% Fiducial Parameters (type)
% A jx2 matrix where j is the number of fiducials. The 2

```

```

% values correspond to an (x,y) coordinate system, where x =
% j(x)*size of images
% M, N, Size1, Size2
%
% OUTPUT: Cropping Data (crop_Data)
% A 2x2 matrix with [xmin, ymin; xmax, ymax]. The minimums
% and maximums are found using the drill holes (ie bone with
% drill hole is useless, so should be cropped out). If no
% drill hole is present on an edge, then 88% of the image is
% used in that direction.
%=====
cropdata = zeros(2, 2);
cropdata(2,2) = round(m*(imgsize(1)*.92));
cropdata(2,1) = round(n*(imgsize(2)*.92));
for i=1:(size(type,1))
newtype = type(i,:);
if newtype(1) == 0 % fiducial at top of image
cropdata(1,2) = refholes(i).subs(2,1); % change cropdata ymin to subs ymax
end
if newtype(1) == m % fiducial equal to number of rows (bottom)
cropdata(2,2) = refholes(i).subs(1,1); % change cropdata ymax to subs ymin
end
if newtype(2) == 0 % fiducial at left of image
cropdata(1,1) = refholes(i).subs(2,2); % change cropdata xmin to subs xmax
end
if newtype(2) == n % fiducial at right of image
cropdata(2,1) = refholes(i).subs(1,2); % change cropdata xmax to subs xmin
end
end
%% Add in aligndata and error check
cropdata(1, 1) = cropdata(1, 1) + aligndata(1, 1) + 1; cropdata(1, 2) = cropdata(1, 2) +
aligndata(1, 2) + 1;
cropdata(2, 1) = cropdata(2, 1) + aligndata(2, 1) + 1; cropdata(2, 2) = cropdata(2, 2) +
aligndata(2, 2) + 1;
if cropdata(1, 1) < 1
cropdata(1, 1) = 1;
end
if cropdata(1, 2) < 1
cropdata(1, 2) = 1;
end
if cropdata(2,1) > round(n*(imgsize(2)*.92));
cropdata(2,1) = round(n*(imgsize(2)*.92));
end
if cropdata(2,2) > round(m*(imgsize(1)*.92));
cropdata(2,2) = round(m*(imgsize(1)*.92));
end
end

```

Fiducial_Selection_2.m

%COPYRIGHT August 2008,

```

% Craig R. Slyfield Jr., Ryan E. Tomlinson
function offmarker = Fiducial_Selection_2(type, ccvalues, m, n)
%=====
% Fidmarkerselect.m
% Added support for non-global variables m, n
%
% This function decides which fiduciary marker (drill hole) should be used
% for Z-alignment based on tiling NCC correlation values
%
%=====
num_Holes = size( type, 1 );
cc_Sum = zeros(num_Holes);
cc_Max = 0;
for j = 1 : num_Holes
if type(j,2) == 0
% Along left side
if type(j,1) == 0
% Upper left
cc_Sum(j) = ( ( ccvalues(1, 1) + ccvalues(m + 1, 1) ) / 2 );
elseif type(j,1) == m
% Bottom left
cc_Sum(j) = ( ( ccvalues(m, 1) + ccvalues(m + 1, m - 1) ) / 2 );
else
% Mid left
if type(j,1) == 1 % First horiz. interface
cc_Sum(j) = ( ( ccvalues( m + 1, 2 ) + ccvalues( 1, 1 ) + ccvalues( 2, 1 ) ) / 3 );
elseif type(j,1) == ( m - 1 ) % Last horiz. interface
cc_Sum(j) = ( ( ccvalues( m + 1, m - 2 ) + ccvalues( m, 1 ) + ccvalues( m - 1, 1 ) ) / 3 );
else
cc_Sum(j) = ( ( ccvalues( m + 1, type(j,1) - 1 ) + ccvalues( m + 1, type(j,1) + 1 ) + ccvalues(
type(j,1), 1 ) + ccvalues( type(j,1) + 1, 1 ) ) / 4 );
end
end
elseif type(j,2) == n
% Along right side
if type(j,1) == 0
% Upper right
cc_Sum(j) = ( ( ccvalues(1, n - 1) + ccvalues(m + 1, 1) ) / 2 );
elseif type(j,1) == m
% Bottom right
cc_Sum(j) = ( ( ccvalues(m - 1, n - 1) + ccvalues(m + 1, m - 1) ) / 2 );
else
% Mid right
if type(j,1) == 1 % First horiz. interface
cc_Sum(j) = ( ( ccvalues( m + 1, 2 ) + ccvalues( 1, n - 1 ) + ccvalues( 2, n - 1 ) ) / 3 );
elseif type(j,1) == ( m - 1 ) % Last horiz. interface
cc_Sum(j) = ( ( ccvalues( m + 1, m - 2 ) + ccvalues( m, n - 1 ) + ccvalues( m - 1, n - 1 ) ) / 3 );
else
cc_Sum(j) = ( ( ccvalues( m + 1, type(j,1) - 1 ) + ccvalues( m + 1, type(j,1) + 1 ) + ccvalues(
type(j,1), n - 1 ) + ccvalues( type(j,1) + 1, n - 1 ) ) / 4 );
end
end
end

```

```

end
elseif type(j,1) == 0
% Mid top
if type(j,2) == 1 % First vert. interface
cc_Sum(j) = ( ( 2*ccvalues( m + 1, 1 ) + ccvalues( 1, 2 ) ) / 3 );
elseif type(j,2) == ( n - 1 ) % Last vert. interface
cc_Sum(j) = ( ( 2*ccvalues( m + 1, 1 ) + ccvalues( 1, n - 2 ) ) / 3 );
else
cc_Sum(j) = ( ( 2*ccvalues( m + 1, 1 ) + ccvalues( 1, type(j,2) - 1 ) + ccvalues( 1, type(j,2) + 1 ) )
/ 4 );
end
elseif type(j,1) == m
% Mid bottom
if type(j,2) == 1 % First vert. interface
cc_Sum(j) = ( ( 2*ccvalues( m + 1, m - 1 ) + ccvalues( m, 2 ) ) / 3 );
elseif type(j,2) == ( n - 1 ) % Last vert. interface
cc_Sum(j) = ( ( 2*ccvalues( m + 1, m - 1 ) + ccvalues( m, n - 2 ) ) / 3 );
else
cc_Sum(j) = ( ( 2*ccvalues( m + 1, m - 1 ) + ccvalues( m, type(j,2) - 1 ) + ccvalues( m, type(j,2) +
1 ) ) / 4 );
end
end
if ( cc_Sum(j) > cc_Max )
offmarker = j;
cc_Max = cc_Sum(j);
end
end

```

Fiducial_Selection_3.m

```

%COPYRIGHT August 2008,
% Craig R Slyfield Jr., Ryan E. Tomlinson
function offset = Fiducial_Selection_3( img, ref_Hole_subs, type, thresh_Value, num_Holes, m,
n, r, val)
% This function takes a tiled UV image and finds the fiducial locations,
% based on labeled locations from the user. Based on the value of
% offmarker, which tells the function which drill hole is
% being used based on tiling NCC correlation values, it only finds the
% drill hole needed in order to save time; If offmarker = 0, this tells the
% function to find all of the drill holes (to be used as the reference
% holes).
%
% Inputs: UV image, drill hole types (positions), m, n, offmarker(the drill
% hole to be used for alignment offsets)
%
% Outputs: drill_Hole(image with drill hole to be used for offset, or all
% the drill holes if being used for reference holes)
%
%=====
rs = r * val; %reduced size drill hole radius

```

```

drill_Holes = imresize( img, val );
yy = size( drill_Holes, 1 );
xx = size( drill_Holes, 2 );
ys = round( size(drill_Holes, 1) / m );
xs = round( size(drill_Holes, 2) / n );
if ref_Hole_subs(2, 1) > yy
ref_Hole_subs(2, 1) = yy;
end
if ref_Hole_subs(2, 2) > xx
ref_Hole_subs(2, 2) = xx;
end
%% Segmenting drill hole fiducials
drill_Holes = drill_Holes > thresh_Value;
drill_Holes = bwareaopen(not(drill_Holes), 5000);
drill_Holes = bwareaopen(not(drill_Holes), 5000);
drill_Holes = imdilate(drill_Holes, strel('disk',3));
y_Position = type(:,1)*ys + 1;
x_Position = type(:,2)*xs + 1;
if y_Position > (size(drill_Holes,1)-1)
y_Position = size(drill_Holes,1)-1;
end
if x_Position > (size(drill_Holes,2)-1)
x_Position = size(drill_Holes,2)-1;
end
drill_Holes = imfill(drill_Holes,sub2ind(size(drill_Holes),y_Position,x_Position));
drill_Holes = imerode(drill_Holes, strel('disk',3));
drill_Holes = bwareaopen(not(drill_Holes), 500);
drill_Holes = bwareaopen(not(drill_Holes), 500);
% -----
drill_Holes = imopen(drill_Holes,strel('disk',10));
drill_Holes = imclose(drill_Holes,strel('disk',10));
drill_Holes = bwareaopen(not(drill_Holes),500);
drill_Holes = bwareaopen(not(drill_Holes),500);
objects = bwlabel( drill_Holes );
%% Isolating specific drill hole
x_Pos = xs * type(2); y_Pos = ys * type(1);
if type(1) == 0 %Along top
y_Pos = 1;
if type(2) == 0 %Upper left
x_Pos = 1;
elseif type(2) == n %Upper right
x_Pos = xx;
else %Middle top
x_Pos = xs * type(2);
end
elseif type(1) == m %Along bottom
y_Pos = yy;
if type(2) == 0 %Bottom left
x_Pos = 1;
elseif type(2) == n %Bottom right
x_Pos = xx;

```



```

else %Middle bottom
x_Pos = xs * type(2);
end
elseif type(2) == 0 %Along left
x_Pos = 1;
y_Pos = ys * type(1);
elseif type(2) == n %Along right
x_Pos = xx;
y_Pos = ys * type(1);
end
hole_Number = objects( y_Pos, x_Pos );
if hole_Number ~= 0
[row col] = find( objects == hole_Number );
drill_Hole_subs = [min(row) min(col) ; max(row) max(col)];
elseif hole_Number == 0
temphole = unique( nonzeros( objects(ref_Hole_subs(1,1) : ref_Hole_subs(2,1),
ref_Hole_subs(1,2) : ref_Hole_subs(2,2)) ));
hole_Number = temphole(1);
[row col] = find( objects == hole_Number );
drill_Hole_subs = [min(row) min(col) ; max(row) max(col)];
end
%% Find offset by registering to original fiducial
offset(1) = (ref_Hole_subs(1,1) - drill_Hole_subs(1,1)) + (ref_Hole_subs(2,1) -
drill_Hole_subs(2,1));
offset(2) = (ref_Hole_subs(1,2) - drill_Hole_subs(1,2)) + (ref_Hole_subs(2,2) -
drill_Hole_subs(2,2));

```

Retile.m

```

function tiledimages = Retile( images, imagesCA, imagesXO, offsets, m, n)
% COPYRIGHT August 2008,
% Craig R Slyfield Jr., Ryan E. Tomlinson
% =====
%
% This function performs M by N tiling on UV images and return the tiled
% image and the various offsets used at all the interfaces.
%
% =====
% Tiling of images in rows
y = size( images(:, :, 1), 1 ); x = size( images(:, :, 1), 2 );
if strcmp( class( images ), 'double' )
rowtiled = zeros( y + 10, x*n, m );
else
rowtiled = uint16( zeros( y + 10, x*n, m ) );
end
rowtiledCA = rowtiled; rowtiledXO = rowtiled;
% The offsets in the y direction should sum to no more than 10, and the x
% offsets are always going to make the tiled image smaller than the sum of
% the widths
% Row tiling

```

```

for l = 1 : m
img1 = images( :, :, (l-1)*n + 1 );
img1CA = imagesCA( :, :, (l-1)*n + 1 );
img1XO = imagesXO( :, :, (l-1)*n + 1 );
t = 1; % Index for offset matrix
for j = 2 : n
img2 = images( :, :, (l-1)*n + j );
img2CA = imagesCA( :, :, (l-1)*n + j );
img2XO = imagesXO( :, :, (l-1)*n + j );
% Adjust canvas size to accommodate registration (only necessary in
% some cases)
bigsize = size(img1);
if size( img2, 1 ) > size( img1, 1 )
bigsize( 1, 1 ) = size( img2, 1 );
end
if size( img2, 2 ) > size( img1, 2 )
bigsize( 1, 2 ) = size( img2, 2 );
end
if strcmp( class( images ), 'double' )
img1dum = zeros(bigsize); img2dum = zeros(bigsize);
else
img1dum = uint16(zeros(bigsize)); img2dum = uint16(zeros(bigsize));
end
img1CAdum = img1dum; img1XOdum = img1dum;
img2CAdum = img2dum; img2XOdum = img2dum;
img1dum( 1 : size( img1, 1 ), 1 : size( img1, 2 ) ) = img1;
img1CAdum( 1 : size( img1CA, 1 ), 1 : size( img1CA, 2 ) ) = img1CA;
img1XOdum( 1 : size( img1XO, 1 ), 1 : size( img1XO, 2 ) ) = img1XO;
img2dum( 1 : size( img2, 1 ), 1 : size( img2, 2 ) ) = img2;
img2CAdum( 1 : size( img2CA, 1 ), 1 : size( img2CA, 2 ) ) = img2CA;
img2XOdum( 1 : size( img2XO, 1 ), 1 : size( img2XO, 2 ) ) = img2XO;
img1 = img1dum; img2 = img2dum;
img1CA = img1CAdum; img2CA = img2CAdum;
img1XO = img1XOdum; img2XO = img2XOdum;
offset = offsets( l, t : t + 1 );
row = Tile( img1, img2, offset );
rowCA = Tile( img1CA, img2CA, offset );
rowXO = Tile( img1XO, img2XO, offset );
temprow = row(5,:);
[I J] = find(temprow<1);
try
A = find(J>size(row,2)*.65);
newJ = J(A);
if newJ(1) > size(row,2)*.65
row(:,newJ(1):newJ(end)) = [];
rowCA(:,newJ(1):newJ(end)) = [];
rowXO(:,newJ(1):newJ(end)) = [];
end
catch
end
t = t + 2;

```

```

img1 = row; img1CA = rowCA; img1XO = rowXO;
end
rowtiled( 1 : size( row, 1 ), 1 : size( row, 2 ), 1 ) = row;
rowtiledCA( 1 : size( rowCA, 1 ), 1 : size( rowCA, 2 ), 1 ) = rowCA;
rowtiledXO( 1 : size( rowXO, 1 ), 1 : size( rowXO, 2 ), 1 ) = rowXO;
end
%% Tiling of rows together
row1 = rowtiled( :, :, 1 );
row1CA = rowtiledCA( :, :, 1 ); row1XO = rowtiledXO( :, :, 1 );
p = 0;
for k = size(row1,1) : -1 : size(row1,1) - 100
if max( row1( k, : ) ) == 0
p = k;
end
end
if p ~= 0
row1( p : end, : ) = [];
row1CA( p : end, : ) = []; row1XO( p : end, : ) = [];
end
p = 0;
for k = 1 : 50
if max( row1( k, : ) ) == 0
p = k;
end
end
if p ~= 0
row1( 1 : p, : ) = [];
row1CA( 1 : p, : ) = []; row1XO( 1 : p, : ) = [];
end
% Assignment and cleanup of first row
t = 1; % Index for offsets matrix
for l = 2 : m
row2 = rowtiled( :, :, l );
row2CA = rowtiledCA( :, :, l ); row2XO = rowtiledXO( :, :, l );
p = 0;
for k = size(row2, 1) : -1 : size(row2, 1) - 100
if max( row2( k, : ) ) == 0
p = k;
end
end
if p ~= 0
row2( p : end, : ) = [];
row2CA( p : end, : ) = []; row2XO( p : end, : ) = [];
end
p = 0;
for k = 1 : 50
if max( row2( k, : ) ) == 0
p = k;
end
end
if p ~= 0

```

```

row2( 1 : p, : ) = [];
row2CA( 1 : p, : ) = []; row2XO( 1 : p, : ) = [];
end
% Adjust canvas size to accommodate registration (only necessary in
% some cases)
bigsize = size(row1);
if size( row2, 1 ) > size( row1, 1 )
bigsize( 1, 1 ) = size( row2, 1 );
end
if size( row2, 2 ) > size( row1, 2 )
bigsize( 1, 2 ) = size( row2, 2 );
end
if strcmp( class( row1 ), 'double' )
row1dum = zeros(bigsize); row2dum = zeros(bigsize);
else
row1dum = uint16(zeros(bigsize)); row2dum = uint16(zeros(bigsize));
end
row1CA = row1dum; row1XO = row1dum;
row2CA = row2dum; row2XO = row2dum;
row1dum( 1 : size( row1, 1 ), 1 : size( row1, 2 ) ) = row1;
row1CA( 1 : size( row1CA, 1 ), 1 : size( row1CA, 2 ) ) = row1CA;
row1XO( 1 : size( row1XO, 1 ), 1 : size( row1XO, 2 ) ) = row1XO;
row2dum( 1 : size( row2, 1 ), 1 : size( row2, 2 ) ) = row2;
row2CA( 1 : size( row2CA, 1 ), 1 : size( row2CA, 2 ) ) = row2CA;
row2XO( 1 : size( row2XO, 1 ), 1 : size( row2XO, 2 ) ) = row2XO;
row1 = row1dum; row2 = row2dum;
row1CA = row1CA; row2CA = row2CA;
row1XO = row1XO; row2XO = row2XO;
offset = offsets( m + 1, t : t + 1 );
tiledimg = ( Tile( row1', row2', offset ) )';
tiledCA = ( Tile( row1CA', row2CA', offset ) )';
tiledXO = ( Tile( row1XO', row2XO', offset ) )';
t = t + 2;
if l > 2
pos = size( tiledimg, 1 ) - ( l - 2 ) * y;
ind = 1;
while ind ~= 0
ind = max( tiledimg( pos, : ) );
pos = pos + 1;
end
tiledimg( pos : end, : ) = [];
tiledCA( pos : end, : ) = []; tiledXO( pos : end, : ) = [];
end
% Cleanup of black space necessary for registration
row1 = tiledimg; row1CA = tiledCA; row1XO = tiledXO;
end
%% Final cleanup of tiled images
x = size( tiledimg, 2 ); y = size( tiledimg, 1 );
p = 0;
for k = x : -1 : x - ( x/3 )
if max( tiledimg( :, k ) ) == 0

```

```

p = k;
end
end
if p ~= 0
tiledimg( :, p : end ) = [];
tiledCA( :, p : end ) = []; tiledXO( :, p : end ) = [];
end
p = 0;
for k = y : -1 : y - ( y/3 )
if max( tiledimg( k, : ) ) == 0
p = k;
end
end
if p ~= 0
tiledimg( p : end, : ) = [];
tiledCA( p : end, : ) = []; tiledXO( p : end, : ) = [];
end
tiledimages( :, :, 1 ) = tiledimg;
tiledimages( :, :, 2 ) = tiledCA;
tiledimages( :, :, 3 ) = tiledXO;

```

Pad_Crop.m

```

%COPYRIGHT August 2008,
% Craig R Slyfield Jr., Ryan E. Tomlinson
function new_Images = Pad_Crop( images, align_Offsets, align_Data, crop_Data )
% images = tiled_top;
% align_Offsets = alignoffset(h,:);
% align_Data = aligndata;
% crop_Data = cropdata;
% =====
% Padandcrop.m [ Name changed to Pad_Crop ]
%
% This function pads and crops input images, using input alignment
% data to pad and cropping data to crop. It is meant to be used with a 3D
% matrix images with three signals.
%
% align_offsets should be a 2 column vector with y in the first and x in
% the second; align_data and crop_Data should be a 2x2 matrix with y and x
% minimums in the first row (respectively) and y and x maximums in the
% second
% =====
yy = size(images, 1); xx = size(images, 2); z = size(images, 3);
% Control Run-Away Alignment Data (highly inaccurate)
if abs(align_Offsets(1)) > align_Data(1) && align_Offsets(1) < 0
align_Offsets(1) = -align_Data(1);
end
if abs(align_Offsets(1)) > align_Data(1) && align_Offsets(1) > 0
align_Offsets(1) = align_Data(1);
end

```

```

if abs(aligned_Offsets(2)) > aligned_Data(2) && aligned_Offsets(2) < 0
aligned_Offsets(2) = -aligned_Data(2);
end
if abs(aligned_Offsets(2)) > aligned_Data(2) && aligned_Offsets(2) > 0
aligned_Offsets(2) = aligned_Data(2);
end
% aligned_Data should be positive and large enough to offset any negative values in
% aligned_Offsets. Therefore, these values should all be positive. Pos_2
% should be exactly the length of the image in that direction -1 larger
% than Pos_1
pos_Y1 = aligned_Offsets(1) + aligned_Data(1, 1) + 1;
pos_Y2 = aligned_Offsets(1) + aligned_Data(1, 1) + yy;
pos_X1 = aligned_Offsets(2) + aligned_Data(2, 1) + 1;
pos_X2 = aligned_Offsets(2) + aligned_Data(2, 1) + xx;
%Dummy aligns the image by holding it at the correct positions (determined above) until cropped
dummy = uint16( zeros( aligned_Data(1, 2) + aligned_Data(1, 1) + yy, aligned_Data(2, 2) +
aligned_Data(2, 1) + xx ) );
%New_Image is a canvas the size of the final product
new_Images = uint16(zeros( size(crop_Data(1,2) : crop_Data(2,2), 2), size(crop_Data(1, 1) :
crop_Data(2, 1), 2), z ) );
for i = 1 : z
%Pad
dummy( pos_Y1 : pos_Y2, pos_X1 : pos_X2 ) = images(:, :, i);
%Crop
new_Images(:, :, i) = dummy( crop_Data(1, 2) : crop_Data(2, 2), crop_Data(1, 1) : crop_Data(2,
1) );
end

```

SubSurf_Fluor_Correct.m

```

%COPYRIGHT August 2008,
% Craig R Slyfield Jr.
function images = SubSurf_Fluor_Correct( images, images_Down, scaling_Factor, h_Size,
spread )
% =====
% opcorrect.m
%
% This function corrects for subsurface fluorescence by subtracting the
% next image down after blurring and darkening using specified conditions.
% images_down can be 3D matrices where the images in the third index should
% match.
% =====
for i = 1 : size( images, 3)
images(:, :, i) = images(:, :, i) - scaling_Factor * imfilter( images_Down(:, :, i), fspecial(
'gaussian', h_Size, spread ) );
end

```

Threshold.m

```

%COPYRIGHT August 2008,
% Craig R Slyfield Jr.
function thresh = Threshold( img,T,hsize,its )
% =====
% % Iterative thresholding algorithm, based on that described in Wu, Barba,
% and Gil 2000. The values chosen (for hsize, g0, tau0 and dtau) were
% chosen either due to numbers given in the article or after testing with
% other values. Inputs are the grayscale image and the desired initial
% threshold.
% =====
g0 = hsize * 0.3;
ginf = .5;
% Gaussian function parameter
n0 = 200;
inf = 10;
% Initial threshold
thresh = ( img > T );
% Initial segmentation
s = thresh * -1;
s = s + ~thresh;
% Characteristic segmentation function
tau0 = its;
tau = tau0;
while tau >= 0
g = ginf + ( g0 - ginf ) * exp( -( (tau0 - tau)/tau ) );
n = ninf + ( n0 - ninf ) * exp( -( (tau0 - tau)/tau ) );
wg = fspecial( 'gaussian', hsize, g );
dT = n * imfilter(s, wg, 'replicate', 'conv' );
T = T + dT;
thresh = ( img > T );
s = thresh * -2;
s = s + ~thresh;
tau = tau - 1;
end

```

Illumination_Profiler.m

```

%COPYRIGHT August 2008,
% Craig R Slyfield Jr.
% Find A (coefficients)
function a = Illumination_Profiler (N)
% Returns the coefficients for the background illumination profile
%% Load Image
H = size(N,1); W = size(N,2);
minbg = zeros(4,4,3);
hsize=round(H/10);
wsize=round(W/10);
gb = fspecial('gaussian',[hsize wsize],sqrt(hsize*wsize));
filtN = imfilter(N,gb);
for y=1:4

```

```

for x=1:4
minbg(y,x,1) = min(min(filtN((((y-1)*(H/4))+1):(y*(H/4)),(((x-1)*(W/4))+1):(x*(W/4)))));
[i,j] = find(filtN((((y-1)*(H/4))+1):(y*(H/4)),(((x-1)*(W/4))+1):(x*(W/4)))<(minbg(y,x,1)+1));
minbg(y,x,2) = i(1); minbg(y,x,3) = j(1);
minbg(y,x,2) = minbg(y,x,2) + (((y-1)*(H/4))+1);
minbg(y,x,3) = minbg(y,x,3) + (((x-1)*(W/4))+1);
end
end
gb = fspecial('gaussian',[3 3],7);
minbg(:, :, 1) = imfilter(minbg(:, :, 1),gb);
n=1:4; y = double([minbg(n,1,2); minbg(n,2,2); minbg(n,3,2); minbg(n,4,2)]);
x = double([minbg(n,1,3); minbg(n,2,3); minbg(n,3,3); minbg(n,4,3)]);
z = double([minbg(n,1,1); minbg(n,2,1); minbg(n,3,1); minbg(n,4,1)]);
X = [ones(size(y)) y x y.^2 x.^2 y.*x];
a = X\z;

```

Register.m

```

%COPYRIGHT August 2008,
% Craig R. Slyfield Jr., Ryan E. Tomlinson, Kyle E. Niemeyer
function [offset, max_cc] = Register( img1, img2 )
% =====
% This function takes two images and returns the offset (y,x) by which they
% overlap according to normalized cross-correlation.
% normxcorr2_mex (matlab executable) function, which uses normalized cross
% correlation to find the offsets. The range of the cross-correlation map is % limited based on the
% known machine offset.
% =====
global pixelsize
machineoff = 45;
% X-ROI
val1 = 45 + round( 1.5*( 5 / pixelsize ) );
% This is the largest allowable overlap
img1 = img1( :, size(img1,2) - val1 + 1 : size(img1,2) );
img2 = img2( :, 1 : round(val1/2) );
val3 = size(img2, 1);
img1 = double(img1); img2 = double(img2);
%NORMXCORR2 – matlab function for normalized cross-correlation
if sum( ( img1(:) ) ) > 40 && sum( ( img2(:) ) ) > 40
if strcmp( mexext, 'mexw32' )
cc = normxcorr2_mex( img2, img1, 'full' );
else
cc = normxcorr2( img2, img1 );
end
cc(1 : val3 - round( 1.5*( 5 / pixelsize ) ), :) = 0; cc(val3 + round( 1.5*( 5 / pixelsize ) ) :
size(cc,1), :) = 0;
cc(:, 1 : (size(cc,2) - machineoff) - round( 1.5*( 5 / pixelsize ) ) ) = 0;
cc(:, (size(cc,2) - machineoff) + round( 1.5*( 5 / pixelsize ) ) : size(cc,2) ) = 0;
%Limit the cc map based on previous knowledge of the machine offset and repeatability
[max_cc, imax] = max( abs( cc(:) ) );

```



```

[ypeak, xpeak] = ind2sub( size(cc), imax(1) );
yoff = ( size( img2, 1 ) - ypeak );
xoff = ( size( cc, 2 ) - xpeak );
offset = [yoff xoff];
else
max_cc = 0;
disp('Triggered Dummy Offset');
offset=[3 18];
end

```

Tile.m

```

%COPYRIGHT August 2008,
% Craig R Slyfield Jr., Ryan E. Tomlinson, Kyle E. Niemeyer
function [image] = Tile(img1, img2, offset)
% images must be ready to be fit left to right
% img1 is on the left, img 2 is on the right
if offset(1,1) >= 0 && offset(1,2) >= 0,
%X image correction
img1(: , size(img1,2)-((offset(1,2))/2):end)=[];
img2(: , (1:((offset(1,2))/2)))=[];
%Y image correction
img1((size(img1,1)+1)-((offset(1,1)))):end , :)=[];
img2(1:((offset(1,1)))) , :)=[];
else
%X image correction
img1(: , size(img1,2)-((offset(1,2))/2):end)=[];
img2(: , (1:((offset(1,2))/2)))=[];
%Y image correction
offset(1,1) = -offset(1,1);
img1(1:((offset(1,1)))) , :)=[];
img2((size(img2,1)+1)-((offset(1,1)))):end , :)=[];
end
image = [img1 img2];

```

Determine_Offset.m

```

%COPYRIGHT August 2008,
% Craig R Slyfield Jr., Ryan E. Tomlinson, Kyle E. Niemeyer
function [offset, max_cc] = Determine_Offset( img1, img2, pixelsize )
% =====
% This function takes two images and returns the offset (y,x) by which they
% overlap. The grayscale images are limited to a ROI before being
% edge-detected, converted to 'double' format, and passed to the
% normxcorr2_mex function, which uses normalized cross correlation to find
% the offsets. The range of the cross-correlation map is limited based on
% the known machine offset. If the edge-detection doesn't have enough
% material to give valid results, a dummy offset is given so that a
% different interface can be used.

```

```

%
% Based on machine offset of [0 45] pixels for everything
% needs normxcorr2_mex (MATLAB Executable file)
%
%=====
machineoff = 45;
% X-ROI
% For img2, it is half the size of img1, fitting inside img1
% val1 is the biggest possible overlap
val1 = 45 + round( 1.5*( 5 / pixelsize ) );
img1 = img1( :, size(img1,2) - val1 + 1 : size(img1,2) );
img2 = img2( :, 1 : round(val1/2) );
val3 = size(img2, 1);
img1 = double(img1); img2 = double(img2);
%NORMXCORR2
if sum( ( img1(:) ) ) > 40 && sum( ( img2(:) ) ) > 40
cc = normxcorr2( img2, img1 );
% Removes part of the cross correlation map using knowledge of machine inaccuracy
cc(1 : val3 - round( 1.5*( 5 / pixelsize ) ), :) = 0; cc(val3 + round( 1.5*( 5 / pixelsize ) ) :
size(cc,1), :) = 0;
cc(:, 1 : (size(cc,2) - machineoff) - round( 1.5*( 5 / pixelsize ) ) ) = 0;
cc(:, (size(cc,2) - machineoff) + round( 1.5*( 5 / pixelsize ) ) : size(cc,2) ) = 0;
% Finds the offset for the images
[max_cc, imax] = max( abs( cc(:) ) );
[ypeak, xpeak] = ind2sub( size(cc), imax(1) );
yoff = ( size( img2, 1 ) - ypeak );
xoff = ( size( cc, 2 ) - xpeak );
offset = [yoff xoff];
else
max_cc = 0;
disp('Triggered Dummy Offset');
offset=[0 45];
end

```

Appendix ***.2 Three-Dimensional Bone Remodeling Quantification

Appendix 1.2.1 3D Mineral Apposition Rate

BFR_CJS 11 19 10.tcl

Amira-Script-File

TO RUN, COPY INTO CONSOLE - source "M:/Or.F/scripts/BFR_CJS 05 27 10.txt"

#Remove from previous run

#remove "\$specimen is currently running"

#Create Dummy Arithmetic to Tell user What Specimen is Running

```

time {
remove -all
create HxArithmetic
Arithmetic setLabel "$specimen is currently running region $region for $Observer"

#Num_Close --> amount to close individual labels
#BW         --> noise removal minimum volume
#InterLabelDist --> distance between labels to close XO and CA
#StartS      --> start slice
#EndS        --> end slice

#set specimen O1NM
#set StartS 0
#set EndS 10
#set BW 3e3
#set InterLabelDist 4

#clear
set BW 3000

set name "M:/BFR/$specimen\_region\_Observer\_Current\_Progress.txt"

#set dateStamp [clock format [clock seconds] -format {%m-%d %I;%M;%S %p}]
set dateStamp 10
# image location
    set path "M:/BFR/$specimen/1mm FOV/Resampled/Masked/Closed"

#
~~~~~PARAMETERS~~~~~
~~~

# code input location
    set scriptPath "M:/Or.F/scripts"

# number of dilations
    set Num_Dilate [expr ceil($InterLabelDist/double(2))]

    echo
\n~~~~~
    echo PROCESSING SPECIMEN: $specimen
    echo \n
set Pfile [open $name a]
puts $Pfile "Processing FITC - Thresh for $specimen"
close $Pfile

```

```

# ~~~~~FITC~~~~~
    echo
~~~~~
    set signal FITC
    echo PROCESSING SIGNAL: $signal for $specimen
    set Thresh_Val $Thresh_Val_Ca

    set imageName \_ $specimen\_ \_ $signal\_ OP\_ FOV.tif
    set Pfile [open $name a]
    puts $Pfile "Flexload FITC"
    close $Pfile

    source "$scriptPath/functions/flex_loadBFR.txt"

    set Pfile [open $name a]
    puts $Pfile "BFR Pairing Proc"
    close $Pfile
    source "$scriptPath/functions/BFR_pairing_processing_8_10_2010.txt"

# ~~~~~TRITC~~~~~
    echo
~~~~~
    set signal TRITC
    echo PROCESSING SIGNAL: $signal for $specimen
    set Thresh_Val $Thresh_Val_XO

    set Pfile [open $name a]
    puts $Pfile "Flexload TRITC"
    close $Pfile
    source "$scriptPath/functions/flex_loadBFR.txt"

    set Pfile [open $name a]
    puts $Pfile "BFRPairing Proc TRITC"
    close $Pfile
    source "$scriptPath/functions/BFR_pairing_processing_8_10_2010.txt"

# ~~~~~BFR~~~~~
    echo MORPHOLOGICAL RECONSTRUCTION

    set Pfile [open $name a]
    puts $Pfile "BFR Pairng morphological"
    close $Pfile
    #source "$scriptPath/functions/BFR_pairing_morphologicalCJS 9 15 10.tcl"

```



```

set imageName [glob "$signal*.tif"]
set imageName [lindex $imageName 0]

set initIm "$path/$imageName"
load -tif +box 0 1 0 1 0 1 +mode 2 $initIm
$imageName setLabel test
set imageSize [test getDims]
echo \tX-DIM: [lindex $imageSize 0] \t Y-DIM: [lindex $imageSize 1] \t Z-DIM:
$lices
remove test

```

```

# ~~~~~LOAD IMAGES~~~~~
echo \n-> LOADING $StartS-$EndS...

set images ""

set images [glob "$signal*.tif"]
set images [lrange $images $StartS $EndS]

eval load -tif +box 0 [expr 1.4*([lindex $imageSize 0]-1)] 0 [expr 1.4*([lindex
$imageSize 1]-1)] 0 [expr 1.667*($lices-1)] +mode 2 $images

[lindex $images 0] setLabel $signal

```

BFR_pairing_processing_8_10_2010.tcl

```

# ~~~~~RESAMPLE~~~~~
echo -> Resampling skipped becuase images should already have been resampled!
# echo -> RESAMPLING...
#
# create HxResample
# Resample data connect $signal
# Resample data send
# Resample mode setValue 1
# Resample voxelSize setValues 1.4 1.4 1.667
# Resample create
# remove Resample
# remove $signal
# $signal\Resampled setLabel $signal
#
#
# ~~~~~THRESHOLD~~~~~
echo -> THRESHOLDING...

```

```

create HxArithmetic
Arithmetic inputA connect $signal
Arithmetic resultChannels setValue 0
Arithmetic expr setValue "A>$Thresh_Val"
Arithmetic resultType setValue 0
Arithmetic create
remove $signal
Result setLabel $signal

```

```

#~~~~~EARLY NOISE
REMOVAL~~~~~
#   echo \n-> NOISE REMOVAL...
#   create HxCalculus $signal\_NoiseRemoval
#   $signal\_NoiseRemoval A connect $signal
#   $signal\_NoiseRemoval MatlabBuffer setValue "if exist('A')==1
#   Noise_Removed = false(size(A,1),size(A,2),size(A,3));
#   Noise_Removed(:,,:)=bwareaopen(A,$EarlyBW);
#   Noise_Removed = uint8(Noise_Removed);
#   end;
#   clear A"
##  $signal\_NoiseRemoval Execute enableSnapToggle 1
#   $signal\_NoiseRemoval Execute snap
#   $signal\_NoiseRemoval Execute send
#   remove $signal\_NoiseRemoval
#   remove $signal\Labels
#   remove $signal
#   Noise_Removed setLabel $signal
#
#   echo -> RESETTING VOXEL RATIO...
#   set size [$signal getDims]
#   $signal setBoundingBox 0 [expr 1.4*([index $size 0]-1)] 0 [expr 1.4*([index
$size 1]-1)] 0 [expr 1.667*([index $size 2]-1)]
#

```

```

#~~~~~LABEL
VOXELS~~~~~
echo -> LABELING...

```

```

create HxLabelVoxel LABELS_$signal
LABELS_$signal data connect $signal
LABELS_$signal regions setValue "Background Signal"
LABELS_$signal boundary01 setValue 0
LABELS_$signal create
remove LABELS_$signal

```

\$signal\Labels setLabel \$signal\Labels

```
# ~~~~~CLOSING~~~~~
create HxGiEditor EDIT_$signal
EDIT_$signal attach $signal\Labels
EDIT_$signal selectMaterial2 1
echo \n-> CLOSING WITH STREL OF $Num_Close...
for {set d 1} {$d < $Num_Close} {incr d} {EDIT_$signal growSelection3D}
for {set d 1} {$d < $Num_Close} {incr d} {EDIT_$signal shrinkSelection3D}
EDIT_$signal add 1
EDIT_$signal detach
remove EDIT_$signal
```

```
# ~~~~~NOISE~~~~~
REMOVAL~~~~~
echo \n-> NOISE REMOVAL...
create HxCalculus $signal\_NoiseRemoval
$signal\_NoiseRemoval A connect $signal\Labels
$signal\_NoiseRemoval MatlabBuffer setValue "if exist('A')==1
Noise_Removed = false(size(A,1),size(A,2),size(A,3));
Noise_Removed(:, :, :) = bwareaopen(A, $BW);
Noise_Removed = uint8(Noise_Removed);
end;
clear A"
$signal\_NoiseRemoval Options setValue 0 0
$signal\_NoiseRemoval Options setToggleVisible 0 1
$signal\_NoiseRemoval Options setValue 1 0
$signal\_NoiseRemoval Options setToggleVisible 1 1
$signal\_NoiseRemoval Apply snap 0 1
$signal\_NoiseRemoval fire
$signal\_NoiseRemoval setViewerMask 16383
$signal\_NoiseRemoval select
$signal\_NoiseRemoval setPickable 1
remove $signal\_NoiseRemoval
remove $signal\Labels
remove $signal
Noise_Removed setLabel $signal

echo -> RESETTING VOXEL RATIO...
set size [$signal getDims]
$signal setBoundingBox 0 [expr 1.4*([lindex $size 0]-1)] 0 [expr 1.4*([lindex
$size 1]-1)] 0 [expr 1.667*([lindex $size 2]-1)]
```



```
# ~~~~~NOISE REMOVED
SURFACE~~~~~
echo -> GENERATING NOISE REMOVED SURFACE...
# CREATE SURFACE GEN MODULE
create HxGMC $signal\_Gen
$signal\_Gen data connect $signal
$signal\_Gen smoothing setValue 2
$signal\_Gen options setValue border 0
#$signal\_Gen data send
$signal\_Gen create
$signal.surf setLabel $signal\_NOISE_REMOVED
remove $signal\_Gen
```

BFR_pairing_morphologicalCJS 11 19 10.tcl

```
set MAR [list ]
set Volume [list ]
set FITC_Area [list ]
set TRITC_Area [list ]
set Pair_Area [list ]
set BBox [list ]
set ImproperEvent [list ]
```

```
echo \n-> DILATING FITC & TRITC WITH STREL OF $Num_Dilate...
```

```
# ~~~~~DILATE SIGNALS~~~~~
for {set i 1} {$i <= 2} {incr i} {
    if {$i == 1} {set signal FITC} elseif {$i == 2} {set signal TRITC}

    create HxLabelVoxel LABELS_$signal
    LABELS_$signal data connect $signal
    LABELS_$signal regions setValue "Background Signal"
    LABELS_$signal boundary01 setValue 0
    LABELS_$signal create
    remove LABELS_$signal

    create HxGiEditor EDIT_$signal
    EDIT_$signal attach $signal.Labels
    EDIT_$signal selectMaterial2 1
    for {set d 1} {$d < $Num_Dilate} {incr d} {EDIT_$signal
growSelection3D}
        EDIT_$signal add 1
        EDIT_$signal detach
        remove EDIT_$signal
```

```

    $signal\Labels setLabel $signal\_Dilated
}

```

```

# ~~~~~~ADD SIGNALS~~~~~

```

```

echo \n-> ADDING DILATED FITC & TRITC...
create HxArithmetic
Arithmetic inputA connect FITC_Dilated
Arithmetic inputB connect TRITC_Dilated
Arithmetic expr setValue "A + B"
Arithmetic create
remove Arithmetic
Result setLabel ADDED

```

```

# ~~~~~~SELECT PAIRS~~~~~

```

```

echo \n-> SELECTING PAIRED REGION...
create HxLabelVoxel LABELS_ADDED
LABELS_ADDED data connect ADDED
LABELS_ADDED regions setValue "Background Paired"
LABELS_ADDED boundary01 setValue 1
LABELS_ADDED create
remove LABELS_ADDED

```

```

ADDED setLabel Pairs
ADDED.Labels setLabel Seed

```

```

# ~~~~~~RECONSTRUCTION~~~~~

```

```

echo \n-> RECONSTRUCTION...
create HxCalculus PairingMATLAB
PairingMATLAB A connect Pairs
PairingMATLAB A1 connect Seed
PairingMATLAB MatlabBuffer setValue "

    Paired = imreconstruct(logical(A1),logical(A),26);
    Paired = Paired.*logical(A);
    \[PairedLabeled, NumberOfPairs\] = bwlabeln(Paired);
    PairedLabeled = uint16(PairedLabeled);
    clear A A1 Paired
    clear NumberOfPairs"
PairingMATLAB Apply snap 0 1
PairingMATLAB fire
PairingMATLAB setPickable 1
remove PairingMATLAB

```

```

PairedLabeled setLabel Paired

echo -> Correcting Voxel Resolution...
set size [$signal getDims]
Paired setBoundingBox 0 [expr 1.4*([lindex $size 0]-1)] 0 [expr 1.4*([lindex
$size 1]-1)] 0 [expr 1.667*([lindex $size 2]-1)]

remove Pairs
remove Seed

# ~~~~~~ASSIGN PAIRS + PW~~~~~

#set Pair_Range [NumberOfPairs getRange]
#remove NumberOfPairs
set NumberOfPairs [lindex [Paired getRange] 1]

# Convert signals to 16-bit
for {set j 1} {$j <= 2} {incr j} {
    if {$j == 1} {set signal FITC} elseif {$j == 2} {set signal TRITC}
    create HxCastField CAST_$signal
    CAST_$signal data connect $signal
    CAST_$signal inputInfo send
    CAST_$signal outputType setValue 2
    CAST_$signal create
    remove CAST_$signal
    remove $signal\_Dilated
    remove $signal
    if {$j == 1} {FITC.to-ushort setLabel FITC} elseif {$j == 2} {TRITC.to-
ushort setLabel TRITC}
}

# Error check (if no pairs found)
if {$NumberOfPairs == 0} {
    echo NO PAIRED LABELS!!
} elseif {$NumberOfPairs > 0} {
    echo \n-> $NumberOfPairs TOTAL PAIRS FOUND.
}

set BBox {}

for {set i 1} {$i <= $NumberOfPairs} {incr i} {

echo $i

```

```

for {set jj 1} {$jj <= 2} {incr jj} {
    if {$jj == 1} {set signal FITC} elseif {$jj == 2} {set signal TRITC}

    echo \n-----> PROCESSING PAIR $i of $NumberOfPairs FOR $signal...
    # Label 0 as background; 1 as selected pair; 2 as
    # the rest
    create HxLabelVoxel LABELS
    LABELS data connect Paired
    LABELS regions setValue "s1 s2 s3"
    LABELS boundary01 setValue [expr $i-1]
    LABELS boundary12 setValue [expr $i]
    LABELS create
    remove LABELS

    # Select ONLY the pair corresponding to material 1
    # from above
    create HxGiEditor EDIT
    EDIT attach Paired.Labels
    EDIT selectMaterial2 2
    EDIT subtract 2
    EDIT selectMaterial2 1
    EDIT add 1
    EDIT detach
    remove EDIT
    Paired.Labels setLabel BFRim

    # Dot multiply by original signal
    create HxArithmetic DotMultiply
    DotMultiply inputA connect $signal
    DotMultiply inputB connect BFRim
    DotMultiply expr setValue "A.*B"
    DotMultiply create
    remove DotMultiply
    Result setLabel $signal\_i

#Point wrap to calculation the volume between the paired labels

    # Create surface of single signal
    create HxGMC SURF\_i
    SURF\_i data connect $signal\_i
    # No smoothing = 0 Constrained smoothing = 2
    SURF\_i smoothing setValue 0
    # Add border = 1
    SURF\_i options setValue border 1
    SURF\_i create

```

```

remove SURF_$i

## Calculate volume for each signal
lappend $signal\_VOLUME [$signal\_$_i\surf getVolume]

remove BFRim SurfaceGen

}

#IF VOLUME OF FITC OR TRITC IS TOO LARGE, PW OR SCS WILL
CRASH AMIRA
#TOO LARGE --> GIVE DUMMY VALUES OF 0 FOR VOLUME AND
INVESTIGATE LATER
if {[expr [TRITC\_$_i\surf getNumTriangles]>750000 || [FITC\_$_i\surf
getNumTriangles]>750000]} {
    echo "pw skipped for event $i"
    set BBox [concat $BBox \{[list 0 0 0 0 0 0]\}]
    lappend PAIRED_VOLUME 0
    lappend ImproperEvent 1
    lappend Volume 0
    lappend FITC_Area [FITC\_$_i\surf getArea]
    lappend TRITC_Area [TRITC\_$_i\surf getArea]
    lappend Pair_Area 0
    set BBox [concat $BBox \{[list 0 0 0 0 0 0]\}]
    create HxSurface Pair\_$_i\_Too_Large
    Pair\_$_i\_Too_Large setIconPosition 20 [expr [expr $i-1]*95 +208]

} else {

    #If labels are well-suited, thin them
    for {set jj 1} {$jj <= 2} {incr jj} {
        if {$jj == 1} {set signal FITC} elseif {$jj == 2} {set signal
TRITC}

        set simplify 1
        source
"M:/Or.F/scripts/functions/Thin_Formation_Label_9_19_2010.tcl"
        Thin_Formation_Label_9_19_2010 $signal\_$_i\surf $simplify
$signal $i 0

        #$_signal\_$_i\surf setLabel $_signal\_$_i\surf_FullRes

        #Generate a point wrapped surface around thinned vertices
        source
"M:/Or.F/scripts/functions/Iteratively_Pointwrap_9_19_2010.tcl"

```

```

Iteratively_Pointwrap_9_19_2010    $signal\_Si\_Thinned
$signal\_Si\.surf_FullRes
remove surf PW

#Refine the PW mesh
create HxRemeshSurface
RemeshSurface setLabel RS
RS data connect $signal\_Si\_Thinned_PW
RS objective setValue 0
RS desiredSize setValue 2 100
RS interpolateOrigSurface setValue 0
RS fire
RS remesh snap
RS remesh send
remove RS
remove SurfaceView

#Re-Thin the thinned label (refined mesh)
set simplify 0
source
"M:/Or.F/scripts/functions/Thin_Formation_Label_9_19_2010.tcl"
Thin_Formation_Label_9_19_2010
$signal\_Si\_Thinned_PW.remeshed $simplify $signal $i 1
$signal\_Si\_Thinned setLabel $signal\_Si\_ThinnedAndRemeshed

#Point Wrap the Re-Thinned Remeshed Label

source
"M:/Or.F/scripts/functions/Iteratively_Pointwrap_FullyProc_Label_9_19_2010.tcl"
Iteratively_Pointwrap_FullyProc_Label_9_19_2010
$signal\_Si\_ThinnedAndRemeshed $signal\_Si\.surf_FullRes
remove surf PW

create HxSimplifier
Simplifier attach $signal\_Si\_ThinnedAndRemeshed_PW
Simplifier simplifyParameters setValue 0 [expr
[$signal\_Si\_ThinnedAndRemeshed_PW getArea]/20]
#Simplifier smooth 5 .5
Simplifier fire
Simplifier simplifyAction setValue 0
Simplifier simplifyAction send
Simplifier detach

#Refine the PW mesh

```

```

create HxRemeshSurface
RemeshSurface setLabel RS
RS data connect $signal\_i\_ThinnedAndRemeshed_PW
RS objective setValue 0
RS interpolateOrigSurface setValue 1
RS desiredSize setValue 1
[$signal\_i\_ThinnedAndRemeshed_PW getNumTriangles]
RS desiredSize setValue 2 100
RS fire
RS remesh snap
RS remesh send
remove RS
remove SurfaceView

remove $signal\_i\_ThinnedAndRemeshed_PW
remove $signal\_i\_Thinned_PW
remove tmp
remove $signal\_i\.surf      $signal\_i
}

set t []
#Check that formation event is well-suited for traditional MAR measures
for {set jj 1} {$jj <= 2} {incr jj} {
    if {$jj == 1} {set signal FITC} elseif {$jj == 2} {set signal
TRITC}

    #MEASURE THICKENSS
    create HxSurfaceThickness {SurfaceThickness}
    SurfaceThickness data connect
$signal\_i\_ThinnedAndRemeshed_PW.remeshed
    SurfaceThickness fire
    SurfaceThickness material setIndex 0 0
    SurfaceThickness action snap 0 1
    SurfaceThickness fire
    $signal\_i\_ThinnedAndRemeshed_PW.Thickness setLabel
thickness

    remove SurfaceThickness

    create HxHistogram
    Histogram data connect thickness
    Histogram fire
    Histogram range setValue 1 0
    Histogram action snap
    Histogram action send
    lappend t [Histogram getMeanInRange]
    remove Histogram thickness

```

```

    }
    set t [lindex [lsort -real $t] 1]

    if {[expr $t < 10]} {

        #Merge two labels
        FITC_${i}_ThinnedAndRemeshed_PW.remeshed duplicate
        FITC_${i}_ThinnedAndRemeshed_PW2.remeshed merge
        TRITC_${i}_ThinnedAndRemeshed_PW.remeshed
        FITC_${i}_ThinnedAndRemeshed_PW2.remeshed setLabel
        Pair_${i}_Thinned

        #PointWrap the two labels with ProbeRad max
        Pair_${i}_Thinned duplicate
        source
        "M:/Or.F/scripts/functions/Iteratively_Pointwrap_FullyProc_Label_9_19_2010.tcl"
        Iteratively_Pointwrap_FullyProc_Label_9_19_2010 Pair_${i}_Thinned
        Pair_${i}_Thinned2
        remove Pair_${i}_Thinned2
        surf setLabel Pair_${i}_Thinned

        #Remesh the surface to fix any errors
        create HxRemeshSurface
        RemeshSurface setLabel RS
        RS data connect Pair_${i}_Thinned_PW
        RS objective setValue 0
        RS interpolateOrigSurface setValue 1
        RS desiredSize setValue 1 [Pair_${i}_Thinned getNumTriangles]
        RS desiredSize setValue 2 100
        RS fire
        RS remesh snap
        RS remesh send
        remove RS
        remove SurfaceView
        remove tmp

        #PointWrap the two labels with ProbeRad max
        Pair_${i}_Thinned_PW.remeshed duplicate
        source
        "M:/Or.F/scripts/functions/Iteratively_Pointwrap_FullyProc_Label_9_19_2010.tcl"
        Iteratively_Pointwrap_FullyProc_Label_9_19_2010
        Pair_${i}_Thinned_PW.remeshed Pair_${i}_Thinned_PW2.remeshed
        remove Pair_${i}_Thinned_PW2.remeshed
        remove Pair_${i}_Thinned_PW
    }

```



```

surf setLabel Pair_${i}_Thinned_PW.remeshed

#Successful MAR measurement
lappend ImproperEvent 0
echo \n-----> PW on pair $i of $NumberOfPairs ... specimen $specimen

set Pfile [open $name a]
puts $Pfile "add fitc and tritc "
close $Pfile

set BBox [concat $BBox \{ [Pair_${i}_Thinned_PW.remeshed_PW
getBoundingBox] \}]
lappend Volume [Pair_${i}_Thinned_PW.remeshed_PW getVolume]
lappend FITC_Area [FITC_${i}_ThinnedAndRemeshed_PW.remeshed
getArea]
lappend TRITC_Area [TRITC_${i}_ThinnedAndRemeshed_PW.remeshed
getArea]
lappend Pair_Area [Pair_${i}_Thinned_PW.remeshed_PW getArea]

Pair_${i}_Thinned_PW.remeshed setIconPosition 20 [expr [expr $i-1]*95
+208]
Pair_${i}_Thinned_PW.remeshed_PW setIconPosition 238 [expr [expr $i-
1]*95 +208]
FITC_${i}.surf_FullRes setIconPosition 20 [expr [expr $i-1]*95+235]
TRITC_${i}.surf_FullRes setIconPosition 20 [expr [expr $i-1]*95+261]
FITC_${i}_ThinnedAndRemeshed_PW.remeshed setIconPosition 189
[expr [expr $i-1]*95+235]
TRITC_${i}_ThinnedAndRemeshed_PW.remeshed setIconPosition 189
[expr [expr $i-1]*95+261]

} else {

lappend ImproperEvent 1
lappend Volume 0
lappend FITC_Area [FITC_${i}_ThinnedAndRemeshed_PW.remeshed
getArea]
lappend TRITC_Area [TRITC_${i}_ThinnedAndRemeshed_PW.remeshed
getArea]
lappend Pair_Area 0
set BBox [concat $BBox \{[list 0 0 0 0 0 0]\}]
create HxSurface No_Thinned_Pair_For_Event_${i}
No_Thinned_Pair_For_Event_${i} setIconPosition 20 [expr [expr $i-1]*95
+208]
FITC_${i}_ThinnedAndRemeshed_PW.remeshed setIconPosition 189
[expr [expr $i-1]*95+235]

```

```

        TRITC_${i}\_ThinnedAndRemeshed_PW.remeshed setIconPosition 189
[expr [expr ${i}-1]*95+261]
        FITC_${i}\.surf_FullRes setIconPosition 20 [expr [expr ${i}-1]*95+235]
        TRITC_${i}\.surf_FullRes setIconPosition 20 [expr [expr ${i}-1]*95+261]
    }

}

remove Pair_${i}\_
remove FITC_${i} TRITC_${i} FITC_${i}\.surf TRITC_${i}\.surf
remove PW PW1 PW2 PW3 PW4 PW5
#remove FITC_${i}\.surf_FullRes TRITC_${i}\.surf_FullRes
remove Pair_${i}\_Thinned

}

```

Appendix 1.2.1 Single Labeled Formation Events

SLS_CJS 02 10 11.tcl

```

# Amira-Script-File
# TO RUN, COPY INTO CONSOLE - source "M:/Or.F/scripts/BFR_CJS 05 27 10.txt"

#Remove from previous run
#remove "$specimen is currently running"

#Create Dummy Arithmetic to Tell user What Specimen is Running
#Num_Close --> amount to close individual labels
#BW --> noise removal minimum volume
#InterLabelDist --> distance between labels to close XO and CA
#StartS --> start slice
#EndS --> end slice

remove -all

create HxArithmetic
Arithmetic setLabel "$specimen is currently running region $region for $Observer"

set name "M:/SLS/SLS_${specimen}\_${region}\_${Observer}\_Current\_Progress.txt"

#Remove all variables during computation to avoid amira crashes
set hide 1

#set specimen O1NM
#set StartS 0

```

```

#set EndS 10
#set BW 3e3
#set InterLabelDist 4

#clear
set BW 3000

#set dateStamp [clock format [clock seconds] -format {%m-%d %I;%M;%S %p}]
set dateStamp 10
# image location
    set path "M:/BFR/$specimen/1mm FOV/Resampled/Masked/Closed"

# ~~~~~PARAMETERS~~~~~

# code input location
    set scriptPath "M:/Or.F/scripts"

# number of dilations
    set Num_Dilate [expr ceil($InterLabelDist/double(2))]

    echo
\n~~~~~
    echo PROCESSING SPECIMEN: $specimen
    echo \n

# ~~~~~FITC~~~~~
    echo
~~~~~
    set signal FITC
    echo PROCESSING SIGNAL: $signal for $specimen
    set Thresh_Val $Thresh_Val_Ca

    set imageName \_ $specimen\_ \_ $signal\_ OP\_ FOV.tif

    source "$scriptPath/functions/flex_loadBFR.txt"

    source "$scriptPath/functions/SLS_pairing_processing_02_10_2011.tcl"

# ~~~~~TRITC~~~~~
    echo
~~~~~
    set signal TRITC

```

```

echo PROCESSING SIGNAL: $signal for $specimen
set Thresh_Val $Thresh_Val_XO

source "$scriptPath/functions/flex_loadBFR.txt"

source "$scriptPath/functions/SLS_pairing_processing_02_10_2011.tcl"

# ~~~~~SLS~~~~~
echo
~~~~~
echo MORPHOLOGICAL RECONSTRUCTION

#source "$scriptPath/functions/BFR_pairing_morphologicalCJS 9 15 10.tcl"
source "$scriptPath/functions/SLS Morphological Processing CJS 02 22 11.tcl"

SLS_pairing_processing_02_10_2011.tcl

#
~~~~~RESAMPLE~~~~~
echo -> Resampling skipped becuase images should already have been resampled!
# echo -> RESAMPLING...
#
# create HxResample
# Resample data connect $signal
# Resample data send
# Resample mode setValue 1
# Resample voxelSize setValues 1.4 1.4 1.667
# Resample create
# remove Resample
# remove $signal
# $signal\Resampled setLabel $signal
#
#
# ~~~~~THRESHOLD~~~~~
echo -> THRESHOLDING...

create HxArithmetic
Arithmetic inputA connect $signal
Arithmetic resultChannels setValue 0
Arithmetic expr setValue "A>$Thresh_Val"
Arithmetic resultType setValue 0
Arithmetic create
remove $signal

```

```

        Result setLabel $signal\_
        $signal\_ duplicate
        $signal\_2 setLabel $signal\_ No_Morph
        $signal\_ setLabel $signal
# ~~~~~EARLY NOISE
REMOVAL~~~~~
#     echo \n-> NOISE REMOVAL...
#     create HxCALCULUS $signal\_ NoiseRemoval
#     $signal\_ NoiseRemoval A connect $signal
#     $signal\_ NoiseRemoval MatlabBuffer setValue "if exist('A')==1
#     Noise_Removed = false(size(A,1),size(A,2),size(A,3));
#     Noise_Removed(:,:,)=bwareaopen(A,$EarlyBW);
#     Noise_Removed = uint8(Noise_Removed);
#     end;
#     clear A"
##    $signal\_ NoiseRemoval Execute enableSnapToggle 1
#     $signal\_ NoiseRemoval Execute snap
#     $signal\_ NoiseRemoval Execute send
#     remove $signal\_ NoiseRemoval
#     remove $signal\Labels
#     remove $signal
#     Noise_Removed setLabel $signal
#
#     echo -> RESETTING VOXEL RATIO...
#     set size [$signal getDims]
#     $signal setBoundingBox 0 [expr 1.4*([lindex $size 0]-1)] 0 [expr 1.4*([lindex
$size 1]-1)] 0 [expr 1.667*([lindex $size 2]-1)]
#
# ~~~~~LABEL
VOXELS~~~~~
    echo -> LABELING...

    create HxLabelVoxel LABELS_$signal
    LABELS_$signal data connect $signal
    LABELS_$signal regions setValue "Background Signal"
    LABELS_$signal boundary01 setValue 0
    LABELS_$signal create
    remove LABELS_$signal
    $signal\Labels setLabel $signal\Labels

# ~~~~~CLOSING~~~~~
    create HxGiEditor EDIT_$signal
    EDIT_$signal attach $signal\Labels
    EDIT_$signal selectMaterial2 1

```

```

echo \n-> CLOSING WITH STREL OF $Num_Close...
for {set d 1} {$d < $Num_Close} {incr d} {EDIT_$signal growSelection3D}
for {set d 1} {$d < $Num_Close} {incr d} {EDIT_$signal shrinkSelection3D}
EDIT_$signal add 1
EDIT_$signal detach
remove EDIT_$signal

```

```

# ~~~~~NOISE REMOVAL~~~~~

```

```

echo \n-> NOISE REMOVAL...
create HxCALCULUS $signal\_NoiseRemoval
$signal\_NoiseRemoval A connect $signal\Labels
$signal\_NoiseRemoval MatlabBuffer setValue "if exist('A')==1
Noise_Removed = false(size(A,1),size(A,2),size(A,3));
Noise_Removed(:,:,.)=bwareaopen(A,$BW);
Noise_Removed = uint8(Noise_Removed);
end;
clear A"
$signal\_NoiseRemoval Options setValue 0 0
$signal\_NoiseRemoval Options setToggleVisible 0 1
$signal\_NoiseRemoval Options setValue 1 0
$signal\_NoiseRemoval Options setToggleVisible 1 1
$signal\_NoiseRemoval Apply snap 0 1
$signal\_NoiseRemoval fire
$signal\_NoiseRemoval setViewerMask 16383
$signal\_NoiseRemoval select
$signal\_NoiseRemoval setPickable 1
remove $signal\_NoiseRemoval

```

```

remove $signal
Noise_Removed setLabel $signal

```

```

echo -> RESETTING VOXEL RATIO...
set size [$signal getDims]
$signal setBoundingBox 0 [expr 1.4*([lindex $size 0]-1)] 0 [expr 1.4*([lindex
$size 1]-1)] 0 [expr 1.667*([lindex $size 2]-1)]

```

```

# ~~~~~NOISE REMOVED
SURFACE~~~~~

```

```

echo -> GENERATING NOISE REMOVED SURFACE...
# CREATE SURFACE GEN MODULE
create HxGMC $signal\_Gen
$signal\_Gen data connect $signal
$signal\_Gen smoothing setValue 2
$signal\_Gen options setValue border 0
#$signal\_Gen data send
$signal\_Gen create

```

```
$signal\.surf setLabel $signal\_NOISE_REMOVED
remove $signal\_Gen
```

SLS Morphological Processing CJS 02 22 11.tcl

```
set Volume [list ]
set FITC_Area [list ]
set TRITC_Area [list ]
set Pair_Area [list ]
set BBox_FITC [list ]
set A_FITC [list ]
set A_TRITC [list ]
set BBox_TRITC [list ]
set ImproperEvent [list ]
```

```
echo \n-> DILATING FITC & TRITC WITH STREL OF $Num_Dilate...
```

```
# ~~~~~DILATE SIGNALS~~~~~
```

```
for {set i 1} {$i <= 2} {incr i} {
    if {$i == 1} {set signal FITC} elseif {$i == 2} {set signal TRITC}
```

```
    create HxLabelVoxel LABELS_$signal
    LABELS_$signal data connect $signal
    LABELS_$signal regions setValue "Background Signal"
    LABELS_$signal boundary01 setValue 0
    LABELS_$signal create
    remove LABELS_$signal
```

```
    create HxGiEditor EDIT_$signal
    EDIT_$signal attach $signal\.Labels
    EDIT_$signal selectMaterial2 1
    for {set d 1} {$d < $Num_Dilate} {incr d} {EDIT_$signal
```

```
growSelection3D}
```

```
    EDIT_$signal add 1
    EDIT_$signal detach
    remove EDIT_$signal
```

```
    $signal\.Labels setLabel $signal\_Dilated
```

```
}
```

```
# ~~~~~ADD SIGNALS~~~~~
```

```
echo \n-> ADDING DILATED FITC & TRITC...
```

```
create HxArithmetic
Arithmetic inputA connect FITC_Dilated
Arithmetic inputB connect TRITC_Dilated
```

```

Arithmetic expr setValue "A + B"
Arithmetic create
remove Arithmetic
Result setLabel ADDED

```

```

# ~~~~~SELECT PAIRS~~~~~
echo \n-> SELECTING PAIRED REGION...
create HxLabelVoxel LABELS_ADDED
LABELS_ADDED data connect ADDED
LABELS_ADDED regions setValue "Background Paired"
LABELS_ADDED boundary01 setValue 1
LABELS_ADDED create
remove LABELS_ADDED

ADDED setLabel Pairs
ADDED.Labels setLabel Seed

echo \n-> RECONSTRUCTION...
create HxCalculus PairingMATLAB
PairingMATLAB A connect Pairs
PairingMATLAB A1 connect Seed
PairingMATLAB fire
PairingMATLAB Execute setState item 2 1

PairingMATLAB MatlabBuffer setState "

    Paired = imreconstruct(logical(A1),logical(A),26);
    Paired = Paired.*logical(A);
    Paired=uint16(Paired);
    clear A A1
    "

PairingMATLAB Options setValue 0 0
PairingMATLAB Options setToggleVisible 0 1
PairingMATLAB Options setValue 1 0
PairingMATLAB Options setToggleVisible 1 1
PairingMATLAB Apply snap 0 1
PairingMATLAB fire
PairingMATLAB setPickable 1
remove PairingMATLAB

echo -> Correcting Voxel Resolution...
set size [$signal getDims]
Paired setBoundingBox 0 [expr 1.4*([lindex $size 0]-1)] 0 [expr 1.4*([lindex
$size 1]-1)] 0 [expr 1.667*([lindex $size 2]-1)]

```


~~~~~ Measure SL.Ev Attributes ~~~~~

#1. Isolate FITC sL.Evs

remove Seed Pairs

create HxArithmetic

Arithmetic inputA connect FITC

Arithmetic inputB connect Paired

Arithmetic expr setValue "A-B"

Arithmetic create

remove Arithmetic

create HxArithmetic

Arithmetic inputA connect Result

Arithmetic expr setValue "A>0"

Arithmetic create

remove Result

Result2 setLabel \$specimen_FITC_SLS_Region_region

#2. Isolate TRITC SL.Evs

create HxArithmetic

Arithmetic inputA connect TRITC

Arithmetic inputB connect Paired

Arithmetic expr setValue "A-B"

Arithmetic create

remove Arithmetic

create HxArithmetic

Arithmetic inputA connect Result

Arithmetic expr setValue "A>0"

Arithmetic create

remove Result

Result2 setLabel \$specimen_TRITC_SLS_Region_region

#3. Count N of single labeled events

echo \n-> RECONSTRUCTION...

create HxCalculus SLS_ID_MATLAB

SLS_ID_MATLAB A connect \$specimen_FITC_SLS_Region_region

SLS_ID_MATLAB A1 connect \$specimen_TRITC_SLS_Region_region

SLS_ID_MATLAB MatlabBuffer setValue "

```

\FITC_S_Evs, N_FITC_Evs\] = bwlabeln(A);
\TRITC_S_Evs, N_TRITC_Evs\] = bwlabeln(A1);
clear A A1 Paired
FITC_S_Evs = uint16(FITC_S_Evs);
TRITC_S_Evs = uint16(TRITC_S_Evs);"
SLS_ID_MATLAB Apply snap 0 1
SLS_ID_MATLAB fire
SLS_ID_MATLAB setPickable 1
remove SLS_ID_MATLAB

#Restore voxel size
echo -> Correcting Voxel Resolution...
$specimen\_FITC_SLS_Region_$region setBoundingBox 0 [expr 1.4*([lindex
$size 0]-1)] 0 [expr 1.4*([lindex $size 1]-1)] 0 [expr 1.667*([lindex $size 2]-1)]
$specimen\_TRITC_SLS_Region_$region setBoundingBox 0 [expr 1.4*([lindex
$size 0]-1)] 0 [expr 1.4*([lindex $size 1]-1)] 0 [expr 1.667*([lindex $size 2]-1)]
FITC_S_Evs setBoundingBox 0 [expr 1.4*([lindex $size 0]-1)] 0 [expr
1.4*([lindex $size 1]-1)] 0 [expr 1.667*([lindex $size 2]-1)]
TRITC_S_Evs setBoundingBox 0 [expr 1.4*([lindex $size 0]-1)] 0 [expr
1.4*([lindex $size 1]-1)] 0 [expr 1.667*([lindex $size 2]-1)]

#Number of Single Label Events in FITC and TRITC
set N_Sl_Ev_FITC [lindex [FITC_S_Evs getRange 1] 1]
set N_Sl_Ev_TRITC [lindex [TRITC_S_Evs getRange 1] 1]
remove N_FITC_Evs2 N_TRITC_Evs2

#~~~~~Surface Area and Bounding Box of each Sl.Ev~~~~~

set color_FITC [list 0 1 0]
set color_TRITC [list 1 .5 .2]

for {set jj 1} {$jj <= 2} {incr jj} {

    if {$jj == 1} {set signal FITC} elseif {$jj == 2} {set signal TRITC}

    create HxCastField CAST_$signal
    CAST_$signal data connect $signal\_S_Evs
    CAST_$signal inputInfo send
    CAST_$signal outputType setValue 2
    CAST_$signal create
    remove CAST_$signal
    remove $signal\_S_Evs
    $signal\_S_Evs.to-ushort setLabel $signal\_S_Evs

```

```

for {set i 1} {$i <= [expr \N_Sl_Ev_$signal]} {incr i} {

    echo \n-----> PROCESSING PAIR $i of [expr \N_Sl_Ev_$signal]

FOR $signal...
    # Label 0 as background; 1 as selected pair; 2 as

    set Pfile [open $name a]
    puts $Pfile "PROCESSING PAIR $i of [expr \N_Sl_Ev_$signal] FOR $signal..."
    close $Pfile

    # the rest
    create HxLabelVoxel LABELS
    LABELS data connect $signal\_S_Evs
    LABELS regions setValue "s1 s2 s3"
    LABELS boundary01 setValue [expr $i-1]
    LABELS boundary12 setValue [expr $i]
    LABELS create
    remove LABELS

    # Select ONLY the pair corresponding to material 1
    # from above
    create HxGiEditor EDIT
    EDIT attach $signal\_S_Evs.Labels
    EDIT selectMaterial2 2
    EDIT subtract 2
    EDIT selectMaterial2 1
    EDIT add 1
    EDIT detach
    remove EDIT

    $signal\_S_Evs.Labels setLabel $signal\_ $i

    # Dot multiply by original signal
    #create HxArithmetic DotMultiply
    #DotMultiply inputA connect $signal\_S_Evs.Labels
    #DotMultiply expr setValue "A./$i"
    #DotMultiply create
    #remove DotMultiply
    #Result setLabel $signal\_ $i

    #Point wrap to calculation the volume between the paired labels

```

```

# Create surface of single signal
create HxGMC SURF_$i
SURF_$i data connect $signal\_ $i
# No smoothing = 0 Constrained smoothing = 2
SURF_$i smoothing setValue 0
# Add border = 1
SURF_$i options setValue border 1
SURF_$i create
remove SURF_$i

## Calculate volume for each signal
lappend $signal\_ VOLUME [$signal\_ $i\ .surf getVolume]
remove $signal\_ S_Evs.Labels
remove BFRim SurfaceGen

#IF VOLUME OF FITC OR TRITC IS TOO LARGE, PW OR SCS
WILL CRASH AMIRA
#TOO LARGE --> GIVE DUMMY VALUES OF 0 FOR VOLUME
AND INVESTIGATE LATER
if {[expr [$signal\_ $i\ .surf getNumTriangles]>750000 ]} {
    echo "pw skipped for event $i"
    set BBox\_ $signal\_ [concat [expr \ $BBox\_ $signal\_ ] \ {[list 0 0 0 0
0 0]\}}]
    lappend Vol\_ $signal\_ 0

    lappend A\_ $signal\_ [$signal\_ $i\ .surf getArea]
    create HxSurface Pair\_ $i\_ Too\_ Large
    Pair\_ $i\_ Too\_ Large setIconPosition 20 [expr [expr $i-1]*95 +208]

} else {

    #If labels are well-suited, thin them
    set Pfile [open $name a]
    puts $Pfile "Starting to thin FOR $signal..."
    close $Pfile

    set simplify 1
    source
"M:/Or.F/scripts/functions/Thin_Formation_Label_9_19_2010.tcl"
Thin_Formation_Label_9_19_2010 $signal\_ $i\ .surf
$simplify $signal\_ $i 0

    # $signal\_ $i\ .surf setLabel $signal\_ $i\ .surf_FullRes

```

```

#Generate a point wrapped surface around thinned vertices
source
"M:/Or.F/scripts/functions/Iteratively_Pointwrap_9_19_2010.tcl"
Iteratively_Pointwrap_9_19_2010 $signal\_i\_Thinned
$signal\_i\.surf_FullRes
remove surf PW

#Refine the PW mesh
create HxRemeshSurface
RemeshSurface setLabel RS
RS data connect $signal\_i\_Thinned_PW
RS objective setValue 0
RS desiredSize setValue 2 100
RS interpolateOrigSurface setValue 0
RS fire
RS remesh snap
RS remesh send
remove RS
remove SurfaceView

set Pfile [open $name a]
puts $Pfile "Rethin"
close $Pfile

#Re-Thin the thinned label (refined mesh)
set simplify 0
source
"M:/Or.F/scripts/functions/Thin_Formation_Label_9_19_2010.tcl"
Thin_Formation_Label_9_19_2010
$signal\_i\_Thinned_PW.remeshed $simplify $signal $i 1
$signal\_i\_Thinned setLabel
$signal\_i\_ThinnedAndRemeshed

set Pfile [open $name a]
puts $Pfile "PW rethinned"
close $Pfile

#Point Wrap the Re-Thinned Remeshed Label

source
"M:/Or.F/scripts/functions/Iteratively_Pointwrap_FullyProc_Label_9_19_2010.tcl"
Iteratively_Pointwrap_FullyProc_Label_9_19_2010
$signal\_i\_ThinnedAndRemeshed $signal\_i\.surf_FullRes
remove surf PW

```

```

                                set Pfile [open $name a]
puts $Pfile "Simplify"
close $Pfile

                                create HxSimplifier
                                Simplifier attach $signal\_i\_ThinnedAndRemeshed_PW
                                Simplifier simplifyParameters setValue 0 [expr
[$signal\_i\_ThinnedAndRemeshed_PW getArea]/20]
                                #Simplifier smooth 5 .5
                                Simplifier fire
                                Simplifier simplifyAction setValue 0
                                Simplifier simplifyAction send
                                Simplifier detach

                                set Pfile [open $name a]
                                puts $Pfile "Remesh"
                                close $Pfile

                                if {[expr [$signal\_i\_ThinnedAndRemeshed_PW
getNumTriangles] > 2000 ]} {

                                #Refine the PW mesh
                                create HxRemeshSurface
                                RemeshSurface setLabel RS
                                RS data connect
                                $signal\_i\_ThinnedAndRemeshed_PW
                                RS objective setValue 0
                                RS interpolateOrigSurface setValue 1
                                RS desiredSize setValue 1
                                [$signal\_i\_ThinnedAndRemeshed_PW getNumTriangles]
                                RS desiredSize setValue 2 100
                                RS fire
                                RS remesh snap
                                RS remesh send
                                remove RS
                                remove SurfaceView

                                remove $signal\_i\_ThinnedAndRemeshed_PW
                                remove $signal\_i\_Thinned_PW
                                remove tmp
                                remove $signal\_i\.surf      $signal\_i

```

```

    } elseif {[expr [$signal\_i\_ThinnedAndRemeshed_PW
getNumTriangles] > 500 ]} {

        $signal\_i\_ThinnedAndRemeshed_PW refine 1
        #Refine the PW mesh
        create HxRemeshSurface
        RemeshSurface setLabel RS
        RS data connect
$signal\_i\_ThinnedAndRemeshed_PW
        RS objective setValue 0
        RS interpolateOrigSurface setValue 1
        RS desiredSize setValue 1
[$signal\_i\_ThinnedAndRemeshed_PW getNumTriangles]
        RS desiredSize setValue 2 100
        RS fire
        RS remesh snap
        RS remesh send
        remove RS
        remove SurfaceView

        remove $signal\_i\_ThinnedAndRemeshed_PW
        remove $signal\_i\_Thinned_PW
        remove tmp
        remove $signal\_i\.surf      $signal\_i

    } else {

        $signal\_i\_ThinnedAndRemeshed_PW setLabel
$signal\_i\_ThinnedAndRemeshed_PW.remeshed
    }

    $signal\_i\_ThinnedAndRemeshed_PW.remeshed
setColor 0 [expr \${color\_signal}]
    $signal\_i\_ThinnedAndRemeshed_PW.remeshed
setColor 1 [expr \${color\_signal}]

    set t []
    #Check that formation event is well-suited for traditional MAR
measures

    set Pfile [open $name a]

```

```

puts $Pfile "Thickness"
close $Pfile

#MEASURE THICKENSS
create HxSurfaceThickness {SurfaceThickness}
SurfaceThickness data connect
$signal\_$_i\_ThinnedAndRemeshed_PW.remeshed
SurfaceThickness fire
SurfaceThickness material setIndex 0 0
SurfaceThickness action snap 0 1
SurfaceThickness fire
$signal\_$_i\_ThinnedAndRemeshed_PW.Thickness
setLabel thickness

remove SurfaceThickness

set Pfile [open $name a]
puts $Pfile "Thickness Histogram"
close $Pfile

create HxHistogram
Histogram data connect thickness
Histogram fire
Histogram range setValue 1 0
Histogram action snap
Histogram action send
lappend th_$signal [Histogram getMeanInRange]
set t [Histogram getMeanInRange]
remove Histogram thickness

if {[expr $t < 10 ]} {
set Pfile [open $name a]
puts $Pfile "Apply bbox for sucessful thin"
close $Pfile

#Successful Sl.Ev measurement
lappend ImproperEvent$signal 0

lappend A_$signal
[$signal\_$_i\_ThinnedAndRemeshed_PW.remeshed getArea]

```



```

                                set BBox\_ $signal [concat [expr \ $BBox\_ $signal]
\{[$signal\_ $i\_ ThinnedAndRemeshed\_PW.remeshed getBoundingBox]\}]

```

```

                                } else {
                                set Pfile [open $name a]
                                puts $Pfile "Apply bbox for UNSucessful thin"
                                close $Pfile

```

```

                                lappend ImproperEvent$signal 1

```

```

                                lappend A\_ $signal
[$signal\_ $i\_ ThinnedAndRemeshed\_PW.remeshed getArea]
                                set BBox\_ $signal [concat [expr \ $BBox\_ $signal]
\{[$signal\_ $i\_ ThinnedAndRemeshed\_PW.remeshed getBoundingBox]\}]

```

```

                                create HxSurface No\_Thinned\_Pair\_For\_Event\_ $i

```

```

                                }

```

```

                                }

```

```

                                set Pfile [open $name a]
                                puts $Pfile "Remove vars"
                                close $Pfile

```

```

                                remove Pair\_ $i\_
                                remove FITC\_ $i TRITC\_ $i FITC\_ $i\surf TRITC\_ $i\surf
                                remove PW PW1 PW2 PW3 PW4 PW5
                                #remove FITC\_ $i\surf\_FullRes TRITC\_ $i\surf\_FullRes
                                remove Pair\_ $i\_ Thinned

```

```

                                if {$hide == 1} {
                                echo yes
                                remove $signal\_ $i\surf\_FullRes
                                remove $signal\_ $i\_ ThinnedAndRemeshed\_PW.remeshed

```

```

                                }

```

```

                                }

```



```

#SurfaceGen data connect $signal\_$_i
#SurfaceGen fire
#SurfaceGen smoothing setIndex 2
#SurfaceGen options setValue 0 1
#SurfaceGen options setToggleVisible 0 1
#SurfaceGen options setValue 1 0
#SurfaceGen options setToggleVisible 1 1
#SurfaceGen border setValue 0 0
#SurfaceGen border setToggleVisible 0 1
#SurfaceGen border setValue 1 0
#SurfaceGen border setToggleVisible 1 1
#SurfaceGen minEdgeLength setMinMax 0 0 0.800000011920929
#SurfaceGen minEdgeLength setValue 0 0
#SurfaceGen materialList setIndex 0 0
#SurfaceGen fire
#SurfaceGen create

```

```

##Define the number of triangles to simplify to
#set nTris [expr [test.surf getNumPoints]/3]
#create HxRemeshSurface {RemeshSurface}
#RemeshSurface data connect test.surf
#{RemeshSurface} fire
#RemeshSurface advancedOptions setValue 0 0
#RemeshSurface advancedOptions setToggleVisible 0 1
#RemeshSurface objective setIndex 0 0
#RemeshSurface triangleArea setValue 0 5
#RemeshSurface triangleArea setValue 1 3
#RemeshSurface triangleArea setValue 2 3
#RemeshSurface vertexValence setValue 0 1
#RemeshSurface triangleQuality setValue 0 10
#RemeshSurface lloydRelaxation setValue 0 40
#RemeshSurface desiredSize setValue 0 nTris
#RemeshSurface errorThresholds setValue 0 0
#RemeshSurface errorThresholds setValue 1 0
#RemeshSurface densityContrast setValue 0 0
#RemeshSurface densityRange setValue 0 0.0535077340900898
#RemeshSurface densityRange setValue 1 45010.1796875
#RemeshSurface interpolateOrigSurface setValue 0
#RemeshSurface remeshOptions1 setValue 0 0
#RemeshSurface remeshOptions1 setToggleVisible 0 1
#RemeshSurface remeshOptions1 setValue 1 1
#RemeshSurface remeshOptions1 setToggleVisible 1 1
#RemeshSurface surfacePathOptions setValue 0 0
#RemeshSurface surfacePathOptions setToggleVisible 0 1

```

```

#RemeshSurface surfacePathOptions setValue 1 1
#RemeshSurface surfacePathOptions setToggleVisible 1 1
#RemeshSurface remeshOptions2 setValue 0
#RemeshSurface contourLayers setMinMax 0 -2147483648 2147483648
#RemeshSurface contourLayers setValue 0 4
#RemeshSurface compute

##### SImplify the input surface to be thinned

if {[expr $simplify == 1]} {
    echo Simplifying the surface before thinning
    $signal\_ $i\.surf setLabel FullRes
    FullRes duplicate
    FullRes setLabel $signal\_ $i\.surf_FullRes
    FullRes2 setLabel $signal\_ $i\.surf_Simple

    create HxSimplifier
    Simplifier attach $signal\_ $i\.surf_Simple
    Simplifier simplifyParameters setValue 0 [expr 0.075*[$signal\_ $i\.surf_Simple
getNumTriangles]]
    #Simplifier smooth 5 .5
    Simplifier fire
    Simplifier simplifyAction setValue 0
    Simplifier simplifyAction send
    $signal\_ $i\.surf_Simple setLabel $signal\_ $i\.surf
    Simplifier detach
} else {
    $surf setLabel $signal\_ $i\.surf
}

#create HxDisplaySurface {SView}
#SView data connect $signal\_ $i\.surf_Simple
#
#SView fire
#SView drawStyle setSpecularLighting 1
#SView drawStyle setTexture 1
#SView drawStyle setAlphaMode 1
#SView drawStyle setNormalBinding 0
#SView drawStyle setSortingMode 1
#SView drawStyle setLineWidth 0.000000

#MEASURE THICKENSS
create HxSurfaceThickness {SurfaceThickness}

```

```

SurfaceThickness data connect $signal\_i\surf
SurfaceThickness fire
SurfaceThickness material setIndex 0 0
SurfaceThickness action snap 0 1
SurfaceThickness fire
$signal\_i\Thickness setLabel thickness

```

```

#CALCULATE SURFACE NORMALS
create HxSurfaceNormals {SurfaceNormals}
SurfaceNormals Surface connect $signal\_i\surf
SurfaceNormals fire
SurfaceNormals NormalType setValue 0
SurfaceNormals InvertedNormal setValue 0 1
SurfaceNormals InvertedNormal setToggleVisible 0 1
SurfaceNormals CreaseAngle setMinMax -3.14159274101257 3.14159274101257
SurfaceNormals CreaseAngle setButtons 0
SurfaceNormals CreaseAngle setIncrement 0.418879
SurfaceNormals CreaseAngle setValue 0
SurfaceNormals CreaseAngle setSubMinMax -3.14159274101257 3.14159274101257
SurfaceNormals doIt snap 0 1
SurfaceNormals fire
SurfaceNormals select
$signal\_i\surf.vertex-normals setLabel vectors
SurfaceNormals deselect

```

```

#COMPUTE NODE LOCATIONS FOR THINNED SURFACE
set nVerts [list ]
set points [list ]

```

```

set [lindex nVerts 0] [$signal\_i\surf getNumPoints]

```

```

#LineSet
create HxLineSet
LineSet setLabel locs

```

```

#LineSet1
create HxLineSet
LineSet setLabel thick

```

```

create HxLineSet
LineSet setLabel d

```

```

create HxLineSet
LineSet setLabel BBmin
create HxLineSet
LineSet setLabel BBmax

create HxLineSet
LineSet setLabel outliers
outliers addPoint $outlier 0 0

set size [$signal\_ $i\ .surf getBoundingBox]
set xs [expr [lindex $size 1] - [lindex $size 0]]
set ys [expr [lindex $size 3] - [lindex $size 2]]
set zs [expr [lindex $size 5] - [lindex $size 4]]

BBmin addPoint [lindex $size 0] [lindex $size 2] [lindex $size 4]
BBmax addPoint [lindex $size 1] [lindex $size 3] [lindex $size 5]

for {set j 1} {$j < $nVerts} {incr j} {
    #set vertLoc [surf.remeshed getPoint $j]
    #set thickj [thickness getValue $j]
    #set direction [vectors getValue $j]
    set vert [$signal\_ $i\ .surf getPoint $j]

    #set x [expr [expr $xL + [expr [expr $thickj/2]*[lindex $direction 0]]] - [lindex $size
0] ]
    #set y [expr [expr $yL + [expr [expr $thickj/2]*[lindex $direction 1]]] - [lindex $size
2] ]
    # set z [expr [expr $zL + [expr [expr $thickj/2]*[lindex $direction 2]]] - [lindex $size 4]
]

    #lappend points "$x $y $z"

    locs addPoint [lindex $vert 0] [lindex $vert 1] [lindex $vert 2]
    thick addPoint [thickness getValue $j] 0 0
    set q [vectors getValue $j]
    d addPoint [lindex $q 0] [lindex $q 1] [lindex $q 2]
}

```

```
#BBox info
#Lineset2
create HxLineSet
LineSet setLabel imSize
imSize addPoint $xs $ys $zs
```

```
create HxCalculus $signal\_Thin\_Matlab
#$signal\_NoiseRemoval A connect $signal\Labels
```

```
#$signal\_Thin\_Matlab MatlabBuffer setValue "
#clear all"
# $signal\_Thin\_Matlab Apply snap 0 1
#$signal\_Thin\_Matlab fire
#$signal\_Thin\_Matlab setViewerMask 16383
#$signal\_Thin\_Matlab select
#$signal\_Thin\_Matlab setPickable 1
```

```
$signal\_Thin\_Matlab A connect locs
$signal\_Thin\_Matlab A1 connect thick
$signal\_Thin\_Matlab A2 connect imSize
$signal\_Thin\_Matlab A3 connect d
$signal\_Thin\_Matlab A4 connect BBmin
$signal\_Thin\_Matlab A5 connect BBmax
$signal\_Thin\_Matlab A6 connect outliers
```

```
$signal\_Thin\_Matlab MatlabBuffer setValue "
warning off;
xs = round(lineset2.points(1));
ys = round(lineset2.points(2));
zs = round(lineset2.points(3));
```

```
thickness = lineset1.points(1,:);
```

```
vectors = lineset3.points';
points = lineset.points';
```

```
BBoxMin = lineset4.points;
```

```

BBoxMax = lineset5.points;
outliers = lineset6.points;

x= points(:,1) + thickness./2.* vectors(:,1) - BBoxMin(1);
y= points(:,2) + thickness./2.* vectors(:,2) - BBoxMin(2);
z= points(:,3) + thickness./2.* vectors(:,3) - BBoxMin(3);

h=uint8(false(ys, xs, zs));

if outliers(1) == 0
    Q=(mean(thickness) + 3.5*std(thickness));
else
    Q=(mean(thickness) + 1.5*std(thickness));
end

i1= find(thickness > 0 );
i2= find(thickness < Q);

i3= find(x > 0 );
i4= find(y > 0 );
i5= find(z > 0 );
i6=find(x<xs);
i7=find(y<ys);
i8=find(z<zs);

p=(intersect(i8, intersect( i7, intersect(i6, intersect( i5, intersect(i4, intersect(i3,
intersect(i1, i2))))))));

x=(BBoxMax(1)-BBoxMin(1))-x;

thickness = thickness(p);
locs = \[x y z];
locs = round(locs(p,:));

i9 = find(locs(:,1)< xs);
i10 = find(locs(:,2)< ys);
i11 = find(locs(:,3)< zs);
i12 = find(locs(:,1)>0);
i13 = find(locs(:,2)>0);
i14 = find(locs(:,3)>0);

```



```

p2=intersect(i14, intersect(i13, intersect(i12, intersect(i11, intersect(i9, i10))));

locs = locs(p2,:);
locs = locs(1:2:end,:);

if length(locs) == 0
    L=1;
else
    for i = 1:length(locs)
        L(i) = sub2ind( [ys xs zs], locs(i,2), locs(i,1), locs(i,3) );
    end
end

h(L)=1;
h=uint8(h);

clear outliers
difference=[xs-lineset2.points(1) ys-lineset2.points(2) zs-lineset2.points(3)];
"

$signal\_Thin\_Matlab Apply snap 0 1
$signal\_Thin\_Matlab fire
$signal\_Thin\_Matlab setViewerMask 16383
$signal\_Thin\_Matlab select
$signal\_Thin\_Matlab setPickable 1

#h.surf setTransform 1 0 0 0 0 1 0 0 0 0 1 0 [expr [lindex $size 0] - $dist] [expr [lindex
$size 2] + $dist ] [lindex $size 4] 1
h setTransform 0 1 0 0 -1 0 0 0 0 0 1 0 0 0 0 1
h applyTransform

set hsize [h getBoundingBox]
set distx [expr [lindex $hsize 1] - [lindex $hsize 0 ] + [difference getValue 0 0 0]]
set disty [difference getValue 0 1 0]
set distz [difference getValue 0 2 0]

h setTransform 1 0 0 0 0 1 0 0 0 0 1 0 [expr [lindex $size 0] + $distx] [expr [lindex
$size 2] + $disty] [expr [lindex $size 4] + $distz] 1
h applyTransform

#CREATE SURFACE
create HxGMC {SurfaceGenh}
SurfaceGenh data connect h

```

```

SurfaceGenh fire
SurfaceGenh smoothing setIndex 0 0
SurfaceGenh options setValue 0 1
SurfaceGenh options setValue 1 0
SurfaceGenh border setValue 0 1
SurfaceGenh minEdgeLength setValue 0 0
SurfaceGenh materialList setIndex 0 0
SurfaceGenh fire
SurfaceGenh create

remove xs xs2 ys ys2 zs zs2 imSize imSize2 imSize3
remove L2 Q2 L Q
remove thick thick2 thick3 locs locs2 locs3 locs4 locs5 p p2 v v2
remove L Q i i1 i2 i3 i4 i5 i6 i7 i8 i9 i10 i11 i12 i13
remove i14 i15 i16 i17 i18 i19 i20 i21 i22 i23 i24 i25 i26 i27 i28 i29
remove d BBmax BBmin BBoxMin2 BBoxMax2 BBmin2 BBmax2 BBoxMax3
remove BBoxMax BBoxMin points BBmax3 BBmin3 d3 d2
remove imSize SurfaceNormals SurfaceThickness thickness vectors
remove x y z x2 y2 z2 points2 thickness3 vectors2 thickness2 vectors3 p3 p4
remove xs2 ys2 zs2 imSize3 Q2 L2 BBoxMin3 BBoxMax2 zs ys xs
remove imSize2
remove $signal\_Thin\_Matlab
remove difference difference2 SurfaceGen
remove outliers outliers outliers outliers outliers outliers outliers outliers outliers
outliers

for {set iii 1} {$iii<15} {incr iii} {
    remove outliers$iii
}
#set hSurfPos [$signal\_ $i getIconPosition]
#h.surf setIconPosition 200 [lindex $hSurfPos 1]
h.surf setLabel $signal\_ $i\_Thinned

remove h SurfaceGenh
#
# #REMOVE THICK REGIONS (THICKEER THAN 1 VOXEL)
# #Scan Convert Surface to image stack
#
#
#set hideNewModules 0
#create HxScanConvertSurface {ScanConvertSurface}
#ScanConvertSurface data connect h.surf
#ScanConvertSurface fire
#set currentbbox [h.surf getBoundingBox]
#ScanConvertSurface bbox setValue 0 [lindex $currentbbox 0]

```

```

#ScanConvertSurface bbox setValue 1 [index $currentbbox 1]
#ScanConvertSurface bbox setValue 2 [index $currentbbox 2]
#ScanConvertSurface bbox setValue 3 [index $currentbbox 3]
#ScanConvertSurface bbox setValue 4 [index $currentbbox 4]
#ScanConvertSurface bbox setValue 5 [index $currentbbox 5]
#ScanConvertSurface dimensions setValue 0 [expr 1.4* [expr [index $currentbbox 1]-
[index $currentbbox 0]]]
#ScanConvertSurface dimensions setValue 1 [expr 1.4* [expr [index $currentbbox 3]-
[index $currentbbox 2]]]
#ScanConvertSurface dimensions setValue 2 [expr 1.4* [expr [index $currentbbox 5]-
[index $currentbbox 4]]]
#ScanConvertSurface materials setIndex 0 0
#ScanConvertSurface options setValue 0 0
#ScanConvertSurface create
#
#

```

```

#####

```

```

## Create labelfield for scan converted surface of thinned label --

```

To close the gaps

```

#create HxLabelVoxel FindThickParts
#FindThickParts data connect h
#FindThickParts regions setValue "Background Signal"
#FindThickParts boundary01 setValue 0
#FindThickParts create
#remove FindThickParts
#
##Close the thinned label
#create HxGiEditor FindThickPartsMorphological
#FindThickPartsMorphological attach h.Labels
#FindThickPartsMorphological selectMaterial2 1
#for {set dd 1} {$dd < 1} {incr dd}
{FindThickPartsMorphological growSelection3D}
#FindThickPartsMorphological add 1
#FindThickPartsMorphological selectMaterial2 0
#for {set dd 1} {$dd < 3} {incr dd}
{FindThickPartsMorphological growSelection3D}
#FindThickPartsMorphological add 0
#FindThickPartsMorphological detach
#
#remove FindThickPartsMorphological
#$signal\_Thin\_SCS\_$_i.Labels ImageData disconnect
#$signal\_Thin\_SCS\_$_i.Labels setLabel
$signal\_Thin\_SCS\_$_i\_Closed
#

```

```

if {[expr $simplify == 1]} {
    remove $signal\_$_i\surf
}
#
#
#
#
#
#
#
#
#
# echo \n\n Formation Label $signal Thinning\: Success \n\n
# echo Point Wrapped Surface is being generated by Standard Code \n
#
#
#
}

```

Iteratively Pointwrap 9_19_2010.tcl

```

proc Iteratively_Pointwrap_9_19_2010 {SurfaceToBePointWrapped OrigRefSurface} {

    set Areas_ThinPW [list ]

    $SurfaceToBePointWrapped setLabel surf

    create HxPointWrap PW
    PW data connect surf
    PW fire
    # Use 100% of max for probe radius
    set ProbeRadMax [PW probeRadius getMaxValue]
    set ProbeRad $ProbeRadMax
    PW probeRadius setValue $ProbeRad
    # Use x-axis as search axis
    PW searchAxis setValue 0
    PW doIt snap
    PW doIt send
}

```

```

set MinAcceptableArea [expr 0.5*[$OrigRefSurface getArea]]

#If Initial PW failed

    #If the PW does not yet exist
    #SET SEARCH AXIS TO X
    if { [expr [string equal [PW getResult] {surf.surf}] != 1 ] } {
        set CurrentArea 0
        remove PW
        if { [expr $CurrentArea < $MinAcceptableArea] } {

            set axis 0
            remove PW
            while { [expr [expr $CurrentArea < $MinAcceptableArea] &&
[expr $ProbeRad > $ProbeRadMax*0.5] ] } {
                echo $ProbeRad
                create HxPointWrap PW
                PW data connect surf
                PW data send
                # Use 90% of max for probe radius
                set ProbeRad [expr $ProbeRad*0.995]
                PW probeRadius setValue $ProbeRad
                # Use x-axis as search axis
                PW searchAxis setValue $axis
                PW doIt snap
                PW doIt send

                if { [expr [string equal [PW getResult] {surf.surf}] != 1 ] } {
                    set CurrentArea 0
                } else {
                    set CurrentArea [surf.surf getArea]
                }

                remove PW
                remove surf.surf
            }
        }
        #If the PW label exists, but is unreasonably small, continually reduce probe
sphere
    } else {
        set ProbeRad $ProbeRadMax
        remove PW
        set CurrentArea [surf.surf getArea]
        if { [expr $CurrentArea < $MinAcceptableArea] } {
            set axis 0
            remove surf.surf
        }
    }

```

```

        while { [expr [expr $CurrentArea < $MinAcceptableArea]
&& [expr $ProbeRad > $ProbeRadMax*0.5] ] } {
            echo $ProbeRad
            create HxPointWrap PW
            PW data connect surf
            PW data send
            # Use 90% of max for probe radius
            set ProbeRad [expr $ProbeRad*0.995]
            PW probeRadius setValue $ProbeRad
            # Use x-axis as search axis
            PW searchAxis setValue $axis
            PW doIt snap
            PW doIt send

            if {[catch {set CurrentArea [surf.surf getArea]}
errmsg]} {
                set CurrentArea 0
            } else {
                set CurrentArea [surf.surf getArea]
                remove surf.surf
                remove PW
            }
        }
    }
    remove surf.surf
    set axis 0
    #CREATE THE PW TO MAKE SURE IT EXISTS FOR SUBSEQUENT IF
STATEMENT
    create HxPointWrap PW
    PW data connect surf
    PW data send
    PW probeRadius setValue $ProbeRad
    # Use x-axis as search axis
    PW searchAxis setValue $axis
    PW doIt snap
    PW doIt send
    remove PW

    if {[catch {set CurrentArea [surf.surf getArea]} errmsg]} {
        set CurrentArea 0
        create HxSurface
        Surface setLabel $axis
    } else {

```

```

        lappend Areas_ThinPW $CurrentArea
        surf.surf setLabel 0
    }

    remove surf.surf

    set ProbeRad $ProbeRadMax
    #SET SEARCH AXIS TO Y
    if {[catch {set CurrentArea [surf.surf getArea]} errmsg]} {
        set CurrentArea 0
        if { [expr $CurrentArea < $MinAcceptableArea] } {

            set axis 1
            remove surf.surf

            while { [expr [expr $CurrentArea < $MinAcceptableArea]
&& [expr $ProbeRad > $ProbeRadMax*0.5] ] } {
                echo $ProbeRad
                create HxPointWrap PW
                PW data connect surf
                PW data send
                # Use 99% of max for probe radius
                set ProbeRad [expr $ProbeRad*0.995]
                PW probeRadius setValue $ProbeRad
                # Use y-axis as search axis
                PW searchAxis setValue $axis
                PW doIt snap
                PW doIt send
                remove PW

                if {[catch {set CurrentArea [surf.surf getArea]}
errmsg]} {
                    set CurrentArea 0
                } else {
                    set CurrentArea [surf.surf getArea]
                    remove surf.surf
                }
            }
        }

        #If the PW label exists, but is unreasonably small, continually reduce probe
sphere
    } else {

        set CurrentArea [surf.surf getArea]
        remove surf.surf
    }

```

```

    if { [expr $CurrentArea < $MinAcceptableArea] } {
        set axis 1
        remove surf.surf

        while { [expr [expr $CurrentArea < $MinAcceptableArea] && [expr
$ProbeRad > $ProbeRadMax*0.5] ] } {
            echo $ProbeRad
            create HxPointWrap PW
            PW data connect surf
            PW data send
            # Use 90% of max for probe radius
            set ProbeRad [expr $ProbeRad*0.995]
            PW probeRadius setValue $ProbeRad
            # Use x-axis as search axis
            PW searchAxis setValue $axis
            PW doIt snap
            PW doIt send
            remove PW

            if {[catch {set CurrentArea [surf.surf getArea]} errmsg]} {
                set CurrentArea 0
            } else {
                set CurrentArea [surf.surf getArea]
                remove surf.surf
            }

            remove surf.surf
        }
    }
}

remove surf.surf

```

#CREATE THE PW TO MAKE SURE IT EXISTS FOR SUBSEQUENT IF STATEMENT

```

create HxPointWrap PW
PW data connect surf
PW data send
PW probeRadius setValue $ProbeRad
# Use x-axis as search axis
PW searchAxis setValue $axis
PW doIt snap
PW doIt send
remove PW

```



```

if {[catch {set CurrentArea [surf.surf getArea]} errmsg]} {
    set CurrentArea 0
    create HxSurface
    Surface setLabel $axis
} else {
    lappend Areas_ThinPW $CurrentArea
    surf.surf setLabel 1
}

remove surf.surf

set ProbeRad $ProbeRadMax
#SET SEARCH AXIS TO Z
if {[catch {set CurrentArea [surf.surf getArea]} errmsg]} {
    set CurrentArea 0
    if { [expr $CurrentArea < $MinAcceptableArea] } {

        set axis 2
        remove surf.surf

        while { [expr [expr $CurrentArea < $MinAcceptableArea]
&& [expr $ProbeRad > $ProbeRadMax*0.5] ] } {
            echo $ProbeRad
            create HxPointWrap PW
            PW data connect surf
            PW data send
            # Use 99% of max for probe radius
            set ProbeRad [expr $ProbeRad*0.995]
            PW probeRadius setValue $ProbeRad
            # Use y-axis as search axis
            PW searchAxis setValue $axis
            PW doIt snap
            PW doIt send
            remove PW

            if {[catch {set CurrentArea [surf.surf getArea]}
errmsg]} {
                set CurrentArea 0
            } else {
                set CurrentArea [surf.surf getArea]
                remove surf.surf
            }
        }
    }
}

```

```

#If the PW label exists, but is unreasonably small, continually reduce probe
sphere
} else {

set CurrentArea [surf.surf getArea]
remove surf.surf

if { [expr $CurrentArea < $MinAcceptableArea] } {
    set axis 2
    remove surf.surf

    while { [expr [expr $CurrentArea < $MinAcceptableArea] && [expr
$ProbeRad > $ProbeRadMax*0.5] ] } {
        echo $ProbeRad
        create HxPointWrap PW
        PW data connect surf
        PW data send
        # Use 90% of max for probe radius
        set ProbeRad [expr $ProbeRad*0.995]
        PW probeRadius setValue $ProbeRad
        # Use x-axis as search axis
        PW searchAxis setValue $axis
        PW doIt snap
        PW doIt send
        remove PW

        if {[catch {set CurrentArea [surf.surf getArea]} errmsg]} {
            set CurrentArea 0
        } else {
            set CurrentArea [surf.surf getArea]
            remove surf.surf
        }

        remove surf.surf
    }
}
}

```

#CREATE THE PW TO MAKE SURE IT EXISTS FOR SUBSEQUENT IF STATEMENT

```

create HxPointWrap PW
PW data connect surf
PW data send
PW probeRadius setValue $ProbeRad
# Use x-axis as search axis

```

```

PW searchAxis setValue $axis
PW doIt snap
PW doIt send
remove PW

if {[catch {set CurrentArea [surf.surf getArea]} errmsg]} {
    set CurrentArea 0
    create HxSurface
    Surface setLabel $axis
} else {
    lappend Areas_ThinPW $CurrentArea
    surf.surf setLabel 2
}

remove surf.surf

set A [list]
lappend A [0 getArea]
lappend A [1 getArea]
lappend A [2 getArea]

#USE MAXIMUM PW Area as the final PW'd surface
set CurrentPWArea [lindex [lsort -real $A] 2]
set CurrentPWSearchAxis [lsearch $A $CurrentPWArea]

$CurrentPWSearchAxis setLabel $SurfaceToBePointWrapped\_PW

remove 0 1 2
#End of Funciton
}

```

Appendix 1.3 Microdamage Quantification

Appendix 1.3.1 Spatial Correlation of Resorption Cavities and Microdamage

```

DxCorr.tcl
#Amira Script
#Preprocesses and crops bone surfaces for Damage spatial correlations

#INITIALIZE VARIABLES

```

```

#BSEnum_Dilate = number of voxels to dilate (in-plane) bone edge image by (15 works
alright)
#CavSize = typical cavity size (used to avoid overlapping bboxes, um)
#NSitesInvVoi = number of Dx and nonDx sites to investigate
#dcrit = minimum distance (um) between two random points on BS

```

```

remove -all

```

```

set desktopDir "Q:/"

```

```

set scriptDir "$desktopDir\uDx/scripts/"
set funcDir "$desktopDir\uDx/scripts/functions/"
set spatialDir "$desktopDir\uDx/$specimen\ROI/SPATIAL/"
set dataDir "$desktopDir\uDx/DATA/"
set ROIDir "$desktopDir\uDx/$specimen\ROI/"
set specimenDir "$desktopDir\uDx/$specimen/"
set PTDPath "$spatialDir\Closed_PTDImages/"
set EdgePath "$desktopDir\uDx/$specimen\ROI/Edge/"
set ThreshPath "$desktopDir\uDx/$specimen\ROI/Thresh/"
set GrayPath "$desktopDir\uDx/$specimen\ROI/UV/"

```

```

set OrigRes 0.7
set GrayRes [list 0.7 0.7 5]
set DxandThreshRes [list 2.8 2.8 2.5]
#Changed to 5.6 um for damage resolution
set DxandThreshRes [list 5.6 5.6 5]
#Make N sites to investigate (NSitesInv) divisible by 4
#The entire bone will be analyzed in 4 parts
set NSitesInv 40
set NVois 4
set NSitesInvVoi [expr ceil([expr $NSitesInv/$NVois])]
set NSlices [expr 800/$NVois]
set CavSize [expr 100/ 5]
set CavSizeH [expr $CavSize/2]
set dcrit [expr 300/5]

```

```

#max dist to be considered associated with damage
set minAssocDist [expr 10/5]

```

```

#nonDxBbox = list of bboxes of non dx random bone points
#RNum_Dilate = how much to dilate each random point at once

```

```
#RDist = Total dist to dilate a point on the BS (should be integer multiplier of
RNum_Dilate)
set nonDxBBox [list]
set DxBBBox [list]
```

```
#Number of voxels for growing damage event along BS (specify um)
set RNum_Dilate 20
#Number of voxels for growing damage events to crop BS (specify um, Tb.Th)
set DxGrowNumDilate [expr ceil([expr 40/5])]
#Number of voxels to dilate bone edge (BSE) to create thick bone surface edge
set BSENum_Dilate [expr ceil([expr 20/[lindex $DxandThreshRes 0]])]
```

```
#set RDist [expr $RNum_Dilate*4]
set netdir $spatialDir
```

```
#RANDOMIZE NUMBER GENEATOR IN MATLAB
```

```
#First 5 random sorting indices are for dx ran BS's
#Last 5 random sorting indices are for dx ran BS's
while {[catch {[ create HxCalcus RanIDPicker] getLabel} errmsg]} { create
HxCalcus RanIDPicker}
RanIDPicker Execute setValue 2 0
RanIDPicker MatlabBuffer setValue "
tic; rand(round(toc*1e6),1); rand(round(toc*1e6),1);
\[v,p\]=sort(rand(2*$NSitesInv,1));
Dxpicker = p(1:$NSitesInv);
nonDxpicker = p($NSitesInv+1:end);
MatlabExportList = {'Dxpicker' 'nonDxpicker'};
"
RanIDPicker Apply snap 0 1
RanIDPicker fire
remove RanIDPicker
```

```
#Save randomized ID into text file
set outfile [open "$dataDir/$specimen\_BlindingCode.txt" a]
puts $outfile "RandomID\tDamage?"
puts $outfile "_____ \t _____"
for {set i 0} {$i < $NSitesInv} {incr i} {
puts $outfile [expr round([Dxpicker getValue $i 0 0])]\t\tDamageEvent[expr $i +1]}
for {set i 0} {$i < $NSitesInv} {incr i} {
puts $outfile [expr round([nonDxpicker getValue $i 0 0])]\t\tNonDamageEvent[expr
$i+1] }
close $outfile
```

```

nonDxpicker setNoRename 1
nonDxpicker save "AmiraMesh ZIP" "$dataDir$specimen\_nonDxpicker.am"

```

```

Dxpicker setNoRename 1
Dxpicker save "AmiraMesh ZIP" "$dataDir$specimen\_Dxpicker.am"

```

```

#####

```

```

#
#
#
#
#
#
#
#
#
#

```

```

#####          RUN CODE FOR EACH SUBVOLUME OF INTEREST          #####

```

```

for {set VOI 1} {$VOI <= $NVois} {incr VOI} {

```

```

echo "Starting Dx correlation network generator for VOI $VOI of $NVois"

```

```

#Image filename indices to load at 5 um out of plane resolution

```

```

set StartS0p7 [expr [expr $VOI-1]*$NSlices]

```

```

set EndS0p7 [expr $StartS0p7 + $NSlices -1]

```

```

set Nims [expr $EndS0p7 - $StartS0p7 + 1 ]

```

```

#Load UV threshold Images at 0.7um

```

```

cd $ThreshPath

```

```

set imageNameThresh [glob "*Thresh_ROI.tif"]

```

```

set imageNameThresh [lindex $imageNameThresh 0]

```

```

set initIm "$ThreshPath/$imageNameThresh"

```

```

load -tif +box 0 1 0 1 0 1 +mode 2 $initIm

```

```

$imageNameThresh setLabel test

```

```

set imageSize0p7 [test getDims]

```

```

remove test

```

```

set filenamesThresh [glob "*Thresh_ROI.tif"]

```

```

set filenamesThresh2 [lrange $ThreshPath$filenamesThresh $StartS0p7 $EndS0p7]

```

```

set Nims [llength $filenamesThresh2]

```

```

set name [[ eval load -tif +box 0 [expr 0.7*([lindex $imageSize0p7 0]-1)] 0 [expr
0.7*([lindex $imageSize0p7 1]-1)] 0 [expr 5*($Nims-1)] +mode 2 $filenamesThresh2]
setLabel]

```

```

$name setLabel Thresh

```

```
set szgray [Thresh getDims]
remove Thresh
```

```
#Load Thresh Images at 2.8um
cd $ThreshPath
set imageName [glob "*Thresh*5p6*.tif"]
set imageNameThresh [lindex $imageName 0]
set initIm "$ThreshPath/$imageNameThresh"
load -tif +box 0 1 0 1 0 1 +mode 2 $initIm
$imageNameThresh setLabel test
set imageSize5p6 [test getDims]
remove test
set filenames [glob "*Thresh*5p6*.tif"]
set filenamesThresh "$ThreshPath/$filenames"
set filenamesThresh [lrange $filenamesThresh $StartS0p7 $EndS0p7]
set name [[eval load -tif +box 0 [expr 5.6*([lindex $imageSize5p6 0]-1)] 0 [expr
5.6*([lindex $imageSize5p6 1]-1)] 0 [expr [expr 5*($Nims-1)]] +mode 2
$filenamesThresh] getLabel]
$name setLabel Thresh5p6
```

```
cd $EdgePath
set imageName [glob "*Edge*5p6*.tif"]
set imageNameEdge [lindex $imageName 0]
set initIm "$EdgePath/$imageNameEdge"
load -tif +box 0 1 0 1 0 1 +mode 2 $initIm
$imageNameEdge setLabel test
set imageSize5p6 [test getDims]
remove test
set filenames [glob "*Edge*5p6*.tif"]
set filenamesEdge "$EdgePath/$filenames"
set filenamesEdge [lrange $filenamesEdge $StartS0p7 $EndS0p7]
set name [[eval load -tif +box 0 [expr 5.6*([lindex $imageSize5p6 0]-1)] 0 [expr
5.6*([lindex $imageSize5p6 1]-1)] 0 [expr [expr 5*($Nims-1)]] +mode 2
$filenamesEdge] getLabel]
$name setLabel BSE5p6
```

```
#Image filename indices to load at 2.6 um out of plane resolution
set StartS2p8 [expr [expr $VOI-1]*$NSlices*2]
set EndS2p8 [expr $StartS2p8 + [expr $NSlices *2 ] -1]
set Nims [expr $EndS2p8-$StartS2p8 + 1 ]
#Load post test damage (PTD) images and are 2.8um
#PTD images have been binary closed by ~70 um to group nearby damage pixels
```

```

cd $PTDPath
set imageNamePTD [glob "*PTD*.tif"]
set imageNamePTD [lindex $imageNamePTD 0]
set initIm "$PTDPath/$imageNamePTD"
load -tif +box 0 1 0 1 0 1 +mode 2 $initIm
$imageNamePTD setLabel test
set imageSize2p8 [test getDims]
remove test
set filenames [glob "*PTD*.tif"]
set filenamesPTD "$PTDPath/$filenames"
set filenamesPTD [lrange $filenamesPTD $StartS2p8 $EndS2p8]
set Nims [llength $filenamesPTD]
set name [lindex [[ eval load -tif +box 0 [expr 2.8*([lindex $imageSize2p8 0]-1)] 0 [expr
2.8*([lindex $imageSize2p8 1]-1)] 0 [expr [expr 2.5*($Nims-1)]] +mode 2
$filenamesPTD] getLabel] 0]
$name setLabel PTD

```

#noise should already have been removed from PTD images

```

#Load Edge Images at 2.8um
cd $EdgePath
set imageNameEdge [glob "*Edge*2p8*.tif"]
set imageNameEdge [lindex $imageNameEdge 0]
set initIm "$EdgePath/$imageNameEdge"
load -tif +box 0 1 0 1 0 1 +mode 2 $initIm
$imageNameEdge setLabel test
set imageSize [test getDims]
remove test
set filenames [glob "*Edge*2p8*.tif"]
set filenamesEdge "$EdgePath/$filenames"
set filenamesEdge [lrange $filenamesEdge $StartS2p8 $EndS2p8]
set name [[eval load -tif +box 0 [expr 2.8*([lindex $imageSize 0]-1)] 0 [expr 2.8*([lindex
$imageSize 1]-1)] 0 [expr [expr 2.5*($Nims-1)]] +mode 2 $filenamesEdge] getLabel]
$name setLabel BSE

```

```

#Load Thresh Images at 2.8um
cd $ThreshPath
set imageName [glob "*Thresh*2p8*.tif"]
set imageNameThresh [lindex $imageName 0]
set initIm "$ThreshPath/$imageNameThresh"
load -tif +box 0 1 0 1 0 1 +mode 2 $initIm
$imageNameThresh setLabel test
set imageSize [test getDims]
remove test

```



```

set filenames [glob "*Thresh*2p8*.tif"]
set filenamesThresh "$ThreshPath/$filenames"
set filenamesThresh [lrange $filenamesThresh $StartS2p8 $EndS2p8]
set name [[eval load -tif +box 0 [expr 2.8*([lindex $imageSize 0]-1)] 0 [expr 2.8*([lindex $imageSize 1]-1)] 0 [expr [expr 2.5*($Nims-1)]] +mode 2 $filenamesThresh] getLabel]
$name setLabel Thresh2p8

```

```

create HxArithmetic
Arithmetic inputA connect Thresh2p8
Arithmetic expr setValue "A>0"
Arithmetic create
remove Arithmetic Thresh2p8
Result setLabel Thresh2p8

```

```

#Create an eroded version of the edge image
set n [[Thresh2p8 duplicate] getLabel]
while {[catch {[ create HxCalculus PTDNR] getLabel} errmsg]} { create HxCalculus
PTDNR }
PTDNR A connect $n
PTDNR Execute setValue 2 0
PTDNR MatlabBuffer setValue "
s=size(A);
for ii = 1:s(3)
mask(:, :, ii) = imerode(A(:, :, ii), strel('disk', 2));
end
MatlabExportList = {'mask'};
"
PTDNR Apply snap 0 1
PTDNR fire
remove PTDNR $n
set bbox [BSE getBoundingBox]
mask setBoundingBox [lindex $bbox 0] [lindex $bbox 1] [lindex $bbox 2] [lindex $bbox
3] [lindex $bbox 4] [lindex $bbox 5]

```

```

create HxArithmetic
Arithmetic inputA connect mask
Arithmetic expr setValue "A>0"
Arithmetic create
remove Arithmetic mask
Result setLabel mask

```

#TEST THIS PORTION WITH THRESHOLD IMAGES THAT ACTUALY MATCH
PTD

#Remove damage from marrow space and surface staining

create HxArithmetic

Arithmetic inputA connect mask

Arithmetic inputB connect PTD

Arithmetic expr setValue "A*B>0"

Arithmetic create

remove Arithmetic PTD

Result setLabel PTD

#Dilate existing damage

#Label 0 as background; 1 as selected random point; 2 as the rest

create HxLabelVoxel PTDLLabels

PTDLLabels data connect PTD

PTDLLabels regions setValue "s1 s2 s3"

#Grab the largest damage events (those with IDs from matlab)

PTDLLabels fire

PTDLLabels boundary01 setValue 0

PTDLLabels boundary12 setValue 1

PTDLLabels create

remove PTDLLabels

create HxGiEditor PTDDilate

PTDDilate attach PTD.Labels

PTDDilate selectMaterial2 2

PTDDilate subtract 2

PTDDilate selectMaterial2 1

#Dilate until event is one cavity plus minimum association distance larger than original
for {set d 1} {\$d <= 1} {incr d} {PTDDilate growSelection3D}

PTDDilate add 1

PTDDilate detach

remove PTDDilate

PTD.Labels ImageData disconnect

create HxCastField

CastField data connect PTD.Labels

CastField inputInfo send

CastField outputType setValue 0

```

CastField create
remove CastField
remove PTD.Labels

```

#This part may be removed if post porcessing is done to crate and save 5.6 um images

```

create HxResample
Resample data connect PTD.to-byte
Resample data send
Resample filter setValue 1
Resample mode setValue 1
Resample fire
Resample voxelSize setValues 5.6 5.6 5
Resample create
remove Resample PTD.to-byte
PTD.Resampled setLabel PTD5p6

```

```

#Sort damage events from largest to smallest
#Pick 30 largest events to spatially correlate w/ cavities
#Largest damage events are more likely to be associated w/ a cavity
while {[catch {[ create HxCalculus BwLabelnDx ] getLabel} errmsg]} { create
HxCalculus BwLabelnDx }
BwLabelnDx A connect PTD5p6
BwLabelnDx MatlabBuffer setValue "
    tic
    num=0;
    minVol = 300;
while num < $NSitesInvVoi
    minVol=floor(minVol*0.85)
    A2=bwareaopen(A, minVol);
    CC=bwconncomp(A2, 26);
    num = CC.NumObjects;
end

    sA=size(A);
    L=single(zeros(sA));

    for ii = 1:CC.NumObjects
        vol(ii) = length(CC.PixelIdxList{ii});
        L(CC.PixelIdxList{ii})=ii;
        \[y,x,z]= ind2sub(\[sA(1) sA(2) sA(3)\], CC.PixelIdxList{ii});
        dxCentroids(:,ii) = \[mean(x) mean(y) mean(z)\];
    }

```

```

end

\ [v,p\] = sort(vol,'descend');

%Largest to smallest
IDS = p(1:$NSitesInvVoi);

%%Equalized histogram
%IDS = v(round( linspace(1,length(v),$NSitesInvVoi+2) ));
%IDS = IDS (1:$NSitesInvVoi);

DxR=false(sA);
temp = DxR;

for i =1:$NSitesInvVoi
    locs = find(L == IDS(i));
    temp(locs)=i;
    DxR = DxR + temp;
end
DxR = uint16(DxR);
A=uint8(A);
timeToDxLabelandNR=toc;
MatlabExportList = {'DxR', 'num', 'IDS','dxCentroids','A','vol'};
"

BwLabelnDx Apply snap 0 1
BwLabelnDx fire
remove BwLabelnDx IDS2 num2
#DxR is the labeled damage event image
set imageSize [PTD5p6 getDims]
DxR setBoundingBox 0 [expr [lindex $imageSize 0]*5.6] 0 [expr [lindex $imageSize
1]*5.6] 0 [expr [lindex $imageSize 2]*5]
A setBoundingBox 0 [expr [lindex $imageSize 0]*5.6] 0 [expr [lindex $imageSize
1]*5.6] 0 [expr [lindex $imageSize 2]*5]
A setLabel PTD_with_NR
set z [lindex $imageSize 2]

#Save damage event volume to text file
set outfile [open "$dataDir/$specimen\_DamageEventVolumes.txt" a]
for {set i 0} {$i < [expr [lindex [vol getDims] 1]- 1 ] } {incr i} {
puts $outfile "$VOI\t\t[expr ceil( [vol getValue 0 $i 0] )]"
}

```

```
close $outfile
```

```
remove vol
```

```
file mkdir "$netdir/Image Files/"
```

```
#DILATE EACH DAMAGE EVENT BY CAVITY SIZE + MAX ASSOCIATION  
DISTANCE
```

```
for {set t 1} {$t <= $NSitesInvVoi} {incr t} {
```

```
    #select damage event i
```

```
    echo -> Diliating Random Point $t of $NSitesInvVoi...
```

```
    #Label 0 as background; 1 as selected random point; 2 as the rest
```

```
    create HxLabelVoxel DxLabels
```

```
    DxLabels data connect DxR
```

```
    DxLabels regions setValue "s1 s2 s3"
```

```
    #Grab the largest damage events (those with IDs from matlab)
```

```
    DxLabels fire
```

```
    DxLabels boundary01 setValue [expr $t-1]
```

```
    DxLabels boundary12 setValue [expr $t]
```

```
    DxLabels create
```

```
    remove DxLabels
```

```
#Dilate each Dx Event to ensure the bbox hits the bone surface
```

```
create HxGiEditor Dx_isolate$t
```

```
Dx_isolate$t attach DxR.Labels
```

```
Dx_isolate$t selectMaterial2 2
```

```
Dx_isolate$t subtract 2
```

```
Dx_isolate$t selectMaterial2 1
```

```
#Dilate until event is one cavity plus minimum association distance larger than original
```

```
for {set d 1} {$d < $DxGrowNumDilate} {incr d} {Dx_isolate$t growSelection3D}
```

```
Dx_isolate$t add 1
```

```
Dx_isolate$t detach
```

```
remove Dx_isolate$t
```

```
DxR.Labels ImageData disconnect
```

```
DxR.Labels setLabel DxDilate
```

```
    set currind [expr [expr $VOI-1] * $NSitesInvVoi + $t - 1]
```

```
    set curr [expr round([Dxpicker getValue $currind 0 0])]
```

```
    echo $t $curr
```

```
echo matlabin $curr curr and $t t DxBBox
```

```

create HxCastField
CastField data connect DxDilate
CastField outputType setValue 0
CastField create
remove CastField DxDilate
DxDilate.to-byte setLabel DxDilate
#SEND TO MATLAB TO DETERMINE BBOX FOR EACH DAMAGE EVENT
while {[catch {[ create HxCalculus BBoxDx] getLabel} errmsg]} { create HxCalculus
BBoxDx }
    BBoxDx A connect DxDilate
    #BBoxDx A1 connect BSE
    #BBoxDx A2 connect BSEThick
    BBoxDx MatlabBuffer setValue "
sz=size(A);
locs = find(A);
\[y,x,z\] = ind2sub(\[sz(1),sz(2), sz(3)\], locs);
BBox(1) = min(x);
BBox(2) = max(x);
BBox(3) = min(y);
BBox(4) = max(y);
BBox(5) = min(z);
BBox(6) = max(z);
MatlabExportList={'BBox'};
"
    BBoxDx Apply snap 0 1
    BBoxDx fire
    remove BBoxDx BBox2

    lappend dxBBox [list [BBox getValue 0 0 0 ] [BBox getValue 0 1 0 ] [BBox
getValue 0 2 0 ] [BBox getValue 0 3 0 ] [BBox getValue 0 4 0 ] [BBox getValue 0 5 0
]]

    #CROP EACH DAMAGE EVENT AND SAVE A NETWORK WITH GRAYSCALE
    AND BONE SURFACES
    #Find current index for randomized damage ID

    #All cropping coordinates are with respect to the image (determined in matlab; y,x -->
    x,y)
    set DxXmin [expr ceil([BBox getValue 0 2 0]*5.6/0.7)]
    set DxXmax [expr floor([BBox getValue 0 3 0]*5.6/0.7)]
    set DxYmin [expr ceil([BBox getValue 0 0 0]*5.6/0.7)]
    set DxYmax [expr floor([BBox getValue 0 1 0]*5.6/0.7)]

```

```

set DxZmin [expr ceil([BBox getValue 0 4 0]*5/5)]
set DxZmax [expr floor([BBox getValue 0 5 0]*5/5)]

#Expand cropping coordinates to maximum associative distance
set DxXmin [expr $DxXmin - $minAssocDist\0.7]
set DxYmin [expr $DxYmin - $minAssocDist\0.7]
set DxZmin [expr $DxZmin - $minAssocDist\5]
set DxXmax [expr $DxXmax + $minAssocDist\0.7]
set DxYmax [expr $DxYmax + $minAssocDist\0.7]
set DxZmax [expr $DxZmax + $minAssocDist\5]

#Check to make sure the expanded boxes do not go beyond the image boundary
if {[expr $DxXmin < 0]} {set DxXmin 0}
if {[expr $DxYmin < 0]} {set DxYmin 0}
if {[expr $DxZmin < 0]} {set DxZmin 0}
if {[expr $DxXmax >= [lindex $szgray 0]]} {set DxXmax [expr [lindex $szgray 0] - 1] }
if {[expr $DxYmax >= [lindex $szgray 1]]} {set DxYmax [expr [lindex $szgray 1] - 1] }
if {[expr $DxZmax >= [lindex $szgray 2]]} {set DxZmax [expr [lindex $szgray 2] - 1] }

set DxXmin [expr int($DxXmin)]
set DxXmax [expr int($DxXmax)]
set DxYmin [expr int($DxYmin)]
set DxYmax [expr int($DxYmax)]
set DxZmin [expr int($DxZmin)]
set DxZmax [expr int($DxZmax)]

#Save Cropping Coordinates for 2.8 um damage events, Zmin,max are with respect to
the SOI, not global coordinates
set outfile [open "$spatialDir$specimen Dx Bounding Boxes $VOI 2.8 um.txt" a]
puts $outfile
"$t\t\tcurr\t\t$DxXmin\t\t$DxXmax\t\t$DxYmin\t\t$DxYmax\t\t$DxZmin\t\t$DxZmax"
"
close $outfile

#Load in appropriate Grayscale slices
cd $GrayPath
set imageNameGray [glob "*UV_ROI*.tif"]
set imageNameGray [lindex $imageNameGray 0]
set initIm "$GrayPath/$imageNameGray"
load -tif +box 0 1 0 1 0 1 +mode 2 $initIm
$imageNameGray setLabel test

```

```

set imageSize0p7 [test getDims]
remove test
set filenames [glob "*UV_ROI*.tif"]
set filenamesGray "$GrayPath/$filenames"
set GrayS [expr ceil( $StartS0p7 + $DxZmin)]
set GrayE [expr floor($StartS0p7 + $DxZmax)]
set Nims [expr $GrayE - $GrayS + 1 ]
set filenamesGray [lrange $filenamesGray [expr int($GrayS)] [expr int($GrayE)]]
set name [[eval load -tif +box 0 [expr 0.7*([lindex $imageSize0p7 0]-1)] 0 [expr
0.7*([lindex $imageSize0p7 1]-1)] 0 [expr 5*($Nims-1)] +mode 2 $filenamesGray]
getLabel]
$name setLabel Gray
Gray crop $DxXmin $DxXmax $DxYmin $DxYmax 0 [expr $Nims - 1]
Gray setLabel Gray$curr

```

```

#Save Grayscale ROI
Gray$curr save "AmiraMesh ZIP" "$netdir/Image
Files/$specimen\_ $curr\_ Grayscale.am"

```

```

#SAVE FITC CHANNEL TO ENSURE THAT DAMAGE WAS ACTUALLY
PRESENT (SECONDARY CHECK, STILL BLINDED)
#Load in appropriate Grayscale slices
#This portion has been removed to save time (loading images takes too long and the
cropping info is stored in .txt files)

```

```

#CREATE A SURF OF A RANDOM SEED POINT
#This is a dummy to avoid biasing spatial correlation
#create HxUniformLabelField3
#MS-Study setLabel AddPointSurface
#Add (x,y,z) point to

```

```

#Create the network file for the current damage event
set outfile [open "$netdir/$specimen\_BS\_investigate\_net\_ $curr\.hx" a]
puts $outfile "
# Amira Script
remove -all
\[load \"$netdir/Image Files/$specimen\_ $curr\_ Grayscale.am\"`] setLabel Gray\_ $curr
viewer 0 setCameraType 0

```

```

create HxOrthoSlice
OrthoSlice data connect Gray\_ $curr

```



```

OrthoSlice fire
OrthoSlice select
OrthoSlice sliceNumber setValue 0
OrthoSlice linearRange setValues 0 \[expr \[OrthoSlice linearRange getValue
1\]*0.75\]
OrthoSlice fire
viewer 0 viewAll
set curr $curr
"
close $outfile

```

```

remove $curr FITC$curr Gray$curr DamageEvent$t BBox
$specimen\_curr\_Grayscale.am DxDilate AddPointSurface

```

```

#End of Dx site cropping loop
}

```

```

remove num IDS

```

```

#####
#
#
#
#
#
#
#
#
#
#
#
#
##### non Damaged Bone Surface
#####
#Dilate    ALL        damage events and mask against bone surface

```

```

create HxLabelVoxel DxRemoveLABELS

```

```

DxRemoveLABELS data connect PTD_with_NR
DxRemoveLABELS regions setValue "Background Signal"
DxRemoveLABELS boundary01 setValue 0
DxRemoveLABELS create
remove DxRemoveLABELS

```

```

#Get ALL bone surface with Damage
create HxGiEditor DxRemoveDilate
DxRemoveDilate attach PTD_with_NR.Labels
DxRemoveDilate selectMaterial2 1
for {set d 1} {$d < [expr $DxGrowNumDilate + 2]} {incr d} {DxRemoveDilate
growSelection3D}
DxRemoveDilate add 1
DxRemoveDilate detach
remove DxRemoveDilate
#This is the bone surface with damage
PTD_with_NR.Labels setLabel BSwDx5p6
BSwDx5p6 ImageData disconnect

```

```

create HxCastField
CastField data connect BSwDx5p6
CastField inputInfo send
CastField outputType setValue 0
CastField create
remove CastField BSwDx5p6
BSwDx5p6.to-byte setLabel BSwDx5p6

```

```

#Create Non Damaged Bone Surface Edge
create HxArithmetic
Arithmetic inputA connect BSwDx5p6
Arithmetic inputB connect BSE5p6
Arithmetic expr setValue "(B-A)>0"
Arithmetic create
remove Arithmetic Dx.Labels
Result setLabel nonDxBSE5p6

```

```

#Create Damaged Bone Surface Edge
#Commented -- for figures only, not req'd for data
#create HxArithmetic
#Arithmetic inputA connect BSwDx5p6
#Arithmetic inputB connect BSE5p6
#Arithmetic expr setValue "B*A"

```

```
#Arithmetic create
#remove Arithmetic
#Result setLabel DxBSE5p6
```

```
#Get ALL bone surface with Damage
create HxLabelVoxel nonDxDilateLABELS
nonDxDilateLABELS data connect nonDxBSE5p6
nonDxDilateLABELS regions setValue "Background Signal"
nonDxDilateLABELS boundary01 setValue 0
nonDxDilateLABELS create
remove nonDxDilateLABELS
```

```
create HxGiEditor nonDxBSEDilate
nonDxBSEDilate attach nonDxBSE5p6.Labels
nonDxBSEDilate selectMaterial2 1
for {set d 1} {$d < 3} {incr d} {nonDxBSEDilate growSelection3D}
nonDxBSEDilate add 1
nonDxBSEDilate detach
remove nonDxBSEDilate
#This is the bone surface with damage
nonDxBSE5p6.Labels ImageData disconnect
nonDxBSE5p6.Labels setLabel nonDxBSEThick5p6
```

```
create HxCastField
CastField data connect nonDxBSEThick5p6
CastField inputInfo send
CastField outputType setValue 0
CastField create
remove CastField nonDxBSEThick5p6
nonDxBSEThick5p6.to-byte setLabel nonDxBSEThick5p6
```

```
#Remove small nonDX Event surfaces
while {[catch {[ create HxCalculus RemSmallBS ] getLabel} errmsg]} { create
HxCalculus RemSmallBS }
RemSmallBS A connect nonDxBSEThick5p6
RemSmallBS MatlabBuffer setValue "
tic
%To speed up time, very small bs's are removed first
A=uint8(bwareaopen(A, 1000));
```

```

MatlabExportList={'A'};
"
#apply snap 0 0 disables auto refresh
RemSmallBS Apply snap 0 1
RemSmallBS fire
remove RemSmallBS
set box [nonDxBSE5p6 getBoundingBox]
A setBoundingBox [lindex $box 0] [lindex $box 1] [lindex $box 2] [lindex $box 3]
[lindex $box 4] [lindex $box 5]
remove nonDxBSEThick5p6
A setLabel nonDxBSEThick5p6

#Mask nonDxBSE with thick nonDxBSE to find only reasonably sized non Dx BS's on
thin version
create HxArithmetic
Arithmetic inputA connect nonDxBSEThick5p6
Arithmetic inputB connect nonDxBSE5p6
Arithmetic expr setValue "A*B"
Arithmetic create
remove Arithmetic nonDxBSE5p6
Result setLabel nonDxBSE5p6

#CREATE RANDOM POINTS ON RESULT
#Bone surface must be isotropic in res.
#Dcrit is the minimum dist (um) between two non Dx centroids
source "$funcDir/RandomBSPoints_NonDx_6_21_11.tcl"
#Random points on the non Dx bone surface with unique IDs
imf2 setLabel R
#R setBoundingBox 0 [expr $DxandThreshRes*([lindex $size 0]-1)] 0 [expr
$DxandThreshRes*([lindex $size 1]-1)] 0 [expr $DxandThreshRes*([lindex $size 2]-1)]
R setBoundingBox [lindex $box 0] [lindex $box 1] [lindex $box 2] [lindex $box 3]
[lindex $box 4] [lindex $box 5]

source "$funcDir/GrowSurface_6_26_11_nonDx.tcl"
RBS setLabel nonDxRBS

for {set t 1} {$t <= $NSitesInvVoi} {incr t} {
    #Grab dilated surface from nonDxRBS
    echo Grab dilated surface from nonDxRBS $t of [expr int($NSitesInvVoi)]
    create HxLabelVoxel nonDxRBSGrab

```

```

nonDxRBSGrab data connect nonDxRBS
nonDxRBSGrab regions setValue "s1 s2 s3"
nonDxRBSGrab boundary01 setValue [expr $t-1]
nonDxRBSGrab boundary12 setValue [expr $t]
nonDxRBSGrab create
remove nonDxRBSGrab

```

```

create HxGiEditor nonDxRBSGrabIsolate$t
nonDxRBSGrabIsolate$t attach nonDxRBS.Labels
nonDxRBSGrabIsolate$t selectMaterial2 2
nonDxRBSGrabIsolate$t subtract 2
nonDxRBSGrabIsolate$t selectMaterial2 1
nonDxRBSGrabIsolate$t add 1
nonDxRBSGrabIsolate$t detach
remove nonDxRBSGrabIsolate$t
nonDxRBS.Labels ImageData disconnect

```

```

#Find current randomized ID for non Dx event
set currind [expr [expr $VOI-1] * $NSitesInvVoi +$t -1 ]
set curr [expr round([nonDxpicker getValue $currind 0 0])]

```

```

echo $t $curr
nonDxRBS.Labels setLabel $curr

```

```

create HxGMC $curr\surfGen
$curr\surfGen data connect $curr
$curr\surfGen smoothing setValue 0
$curr\surfGen options setValue border 0
$curr\surfGen create
remove $curr\surfGen
$curr\.surf setNoRename 1

```

```

#CHECK THAT BBOX OF EVENT IS SUFFICIENTLY LARGE< IF NOT
# delete variables and try new random non dx bs point

```

```

#$curr\.surf save "HxSurface binary" "$netdir\Image Files/$specimen\_curr\_BS.am"

```

```

create HxLabelVoxel nonDxRGrab
nonDxRGrab data connect R
nonDxRGrab regions setValue "s1 s2 s3"
nonDxRGrab boundary01 setValue [expr $t-1]
nonDxRGrab boundary12 setValue [expr $t]
nonDxRGrab create

```

```
remove nonDxRGrab
```

```
#Create a surf of the seed point  
create HxGiEditor nonDxRGrabIsolate  
nonDxRGrabIsolate attach R.Labels  
nonDxRGrabIsolate selectMaterial2 2  
nonDxRGrabIsolate subtract 2  
nonDxRGrabIsolate selectMaterial2 1  
nonDxRGrabIsolate add 1  
nonDxRGrabIsolate detach  
remove nonDxRGrabIsolate  
R.Labels ImageData disconnect
```

```
create HxGMC $curr\surfGen  
$curr\surfGen data connect R.Labels  
$curr\surfGen smoothing setValue 0  
$curr\surfGen options setValue border 1  
$curr\surfGen create  
remove $curr\surfGen  
$curr\surf setNoRename 1
```

```
R.surf setNoRename 1  
R.surf save "HxSurface binary" "$netdir\Image Files/$specimen\_ $curr\_Seed.am"  
set BBox [$curr\surf getBoundingBox]
```

```
#All cropping coordinates are with respect to the image (determined in matlab; y,x -->  
x,y)
```

```
set nonDxXmin [expr ceil([lindex $BBox 0]/0.7)]  
set nonDxXmax [expr ceil([lindex $BBox 1]/0.7)]  
set nonDxYmin [expr ceil([lindex $BBox 2]/0.7)]  
set nonDxYmax [expr ceil([lindex $BBox 3]/0.7)]  
set nonDxZmin [expr ceil([lindex $BBox 4]/5)]  
set nonDxZmax [expr ceil([lindex $BBox 5]/5)]
```

```
#Expand cropping coordinates to maximum associative distance  
set nonDxXmin [expr $nonDxXmin - $minAssocDist\0.7]  
set nonDxYmin [expr $nonDxYmin - $minAssocDist\0.7]  
set nonDxZmin [expr $nonDxZmin - $minAssocDist\5]  
set nonDxXmax [expr $nonDxXmax + $minAssocDist\0.7]  
set nonDxYmax [expr $nonDxYmax + $minAssocDist\0.7]  
set nonDxZmax [expr $nonDxZmax + $minAssocDist\5]
```

```
#Check to make sure the expanded boxes do not go beyond the image boundary  
if {[expr $nonDxXmin < 0]} {set nonDxXmin 0}  
if {[expr $nonDxYmin < 0]} {set nonDxYmin 0}
```

```

if {[expr $nonDxZmin < 0]} {set nonDxZmin 0}
if {[expr $nonDxXmax > [lindex $szgray 0]]} {set nonDxXmax [lindex $szgray 0]}
if {[expr $nonDxYmax > [lindex $szgray 1]]} {set nonDxYmax [lindex $szgray 1]}
if {[expr $nonDxZmax > [lindex $szgray 2]]} {set nonDxZmax [lindex $szgray 2]}

set nonDxXmin [expr int($nonDxXmin)]
set nonDxXmax [expr int($nonDxXmax)]
set nonDxYmin [expr int($nonDxYmin)]
set nonDxYmax [expr int($nonDxYmax)]
set nonDxZmin [expr int($nonDxZmin)]
set nonDxZmax [expr int($nonDxZmax)]

#Load in appropriate Grayscale slices
cd $GrayPath
set filenames [glob "*UV*ROI.tif"]
set filenamesGray "$GrayPath/$filenames"

set GrayS [expr ceil( $StartS0p7 + $nonDxZmin)]
set GrayE [expr floor($StartS0p7 + $nonDxZmax)]
set Nims [expr $GrayE - $GrayS + 1 ]
set filenamesGray [lrange $filenamesGray [expr int($GrayS)] [expr int($GrayE)]]
set name [[eval load -tif +box 0 [expr 0.7*([lindex $imageSize0p7 0]-1)] 0 [expr
0.7*([lindex $imageSize0p7 1]-1)] 0 [expr 5*($Nims-1)] +mode 2 $filenamesGray]
getLabel]
$name setLabel Gray

Gray crop $nonDxXmin $nonDxXmax $nonDxYmin $nonDxYmax 0 [expr $Nims -1
]
Gray setLabel Gray$curr

#Save Grayscale ROI
Gray$curr setNoRename 1
Gray$curr save "AmiraMesh ZIP" "$netdir/Image
Files/$specimen\_ $curr\_ Grayscale.am"

#Save Cropping Coordinates for 2.8 um non-damage events, Zmin,max are with
respect to the SOI, not global coordinates
set outfile [open "$spatialDir$specimen Non Dx Bounding Boxes $VOI 2.8 um.txt" a]

```

```

    puts $outfile
"$t\t\t$curr\t\t$nonDxXmin\t\t$nonDxXmax\t\t$nonDxYmin\t\t$nonDxYmax\t\t$nonDx
Zmin\t\t$nonDxZmax"
    close $outfile

    remove Gray$curr R.surf R.Labels $curr $curr\.surf

#Create the network file for the current non damage event
#Create the network file for the current damage event
    set outfile [open "$netdir\specimen\_BS_investigate_net_$curr\.hx" a]
    puts $outfile "
# Amira Script
remove -all
\[load \"$netdir\Image Files\specimen\_curr\_Grayscale.am\" \] setLabel Gray_$curr
viewer 0 setCameraType 0

create HxOrthoSlice
OrthoSlice data connect Gray_$curr
OrthoSlice fire
OrthoSlice select
OrthoSlice sliceNumber setValue 0
OrthoSlice linearRange setValues 0 \[expr \[OrthoSlice linearRange getValue
1\]*0.75\]
OrthoSlice fire

viewer 0 viewAll

set curr $curr
"
close $outfile

#End of non Dx site cropping loop
}

remove PTD PTD5p6 BSE BSE5p6 nonDxBSE5p6 BSEThick BSEThick5p6 Thresh
remove BSE Tresh2p8 mask PTDLabels DxR nonDxBSEThick5p6 dxCentroids
nonDxRBS
remove RBS BBox num IDS R Thresh2p8 Thresh5p6 BSwDx5p6 nonDxBSEThick5p2
PTD_with_NR

#End of subvolume loop
}

```


RandomBSPoints_NonDX_6_21_11.tcl

```
while {[catch {[ create HxCalculus PairingMATLAB ] getLabel} errmsg]} { create
HxCalculus PairingMATLAB }
PairingMATLAB A connect nonDxBSE5p6
PairingMATLAB A1 connect dxCentroids
#PairingMATLAB A1 connect dxBBox
PairingMATLAB MatlabBuffer setValue "
imf=A;
clear A
sz = size(imf);
locs=find(imf);

dxBBox=squeeze(A1);

tic; rand(round(toc*1e6),1); rand(round(toc*1e6),1);

p=0;
while size(p,1)<$NSitesInvVoi

R=rand($NSitesInvVoi*10,1);
LR = locs(round(R*length(locs)));

\[yR,xR,zR\]= ind2sub(\[sz(1) sz(2) sz(3)\], LR);

xRmin = find(xR<=$CavSize);
yRmin = find(yR<=$CavSize);
zRmin = find(zR<=$CavSize);
xRmax= find(xR>=sz(2)-$CavSize);
yRmax = find(yR>=sz(1)-$CavSize);
zRmax = find(zR>=sz(3)-$CavSize);
hits = unique(\[xRmin; yRmin; zRmin; xRmax; yRmax; zRmax\]);

p=zeros($NSitesInvVoi,1);
p(hits)=1;
p=find(~p);
p=\[yR(p) xR(p) zR(p)\];

for j=1:$NSitesInvVoi
for i =1:size(p,1)
```

```

d(i,j)=sqrt( (p(i,2)-dxBBox(1,j))^2 +...
            (p(i,1)-dxBBox(2,j))^2 +...
            (p(i,3)-dxBBox(3,j))^2 );

hitsDxBBox (i,j) = d(i,j) <= $dcrit;
end
end

hitsDxBBox = find(~logical(sum(hitsDxBBox')));
p=[p(hitsDxBBox,1) p(hitsDxBBox,2) p(hitsDxBBox,3)];

end

j=1; removed=1; Npoints=100;
newp=p;
while j < Npoints || size(newp,1) < $NSitesInvVoi

    p_curr = newp(j,:);
    p_othersx = \[newp(1:j-1,2); newp(j+1:end,2)];
    p_othersy = \[newp(1:j-1,1); newp(j+1:end,1)];
    p_othersz = \[newp(1:j-1,3); newp(j+1:end,3)];

    p_others = \[p_othersy p_othersx p_othersz];

    cond=[];
    d=[];
    for i = 1:size(p_others,1)
        d(i)=sqrt( (p_curr(1)-p_others(i,1))^2 +...
                  (p_curr(2)-p_others(i,2))^2 +...
                  (p_curr(3)-p_others(i,3))^2);

        cond(i)=d(i)>=$dcrit ;
    end

    newp = \[p_curr; p_others(find(cond), :)];
    if size(newp,1)==size(p_others,1)+1
        removed = 1;
    else
        removed = 0;
    end
end

```

```

Npoints = size(newp,1);

if size(newp,1) < $NSitesInvVoi

p2=0;

while p2 < $NSitesInvVoi
    R2=rand($NSitesInvVoi*10,1);
    LR2 = locs(round(R2*length(locs)));

    [yR2,xR2,zR2]= ind2sub([sz(1) sz(2) sz(3)], LR2);

    xRmin2 = find(xR2<=$CavSize);
    yRmin2 = find(yR2<=$CavSize);
    zRmin2 = find(zR2<=$CavSize);
    xRmax2= find(xR2>=sz(2)-$CavSize);
    yRmax2 = find(yR2>=sz(1)-$CavSize);
    zRmax2 = find(zR2>=sz(3)-$CavSize);

    hits2 = unique([xRmin2; yRmin2; zRmin2; xRmax2; yRmax2; zRmax2]);

    p2=zeros($NSitesInvVoi,1);

    p2(hits2)=1;
    p2=find(~p2);

    p2=[yR2(p2) xR2(p2) zR2(p2)];

end
hitsDxBBox=[];

%for j=1:$NSitesInvVoi

for j=1: size(dxBBox, 2)
    for i =1:size(p2,1)
        d(i,j)=sqrt( (p2(i,2)-dxBBox(1,j))^2 +...
            (p2(i,1)-dxBBox(2,j))^2 +...
            (p2(i,3)-dxBBox(3,j))^2 );

        hitsDxBBox (i,j) = d(i,j) <= $dcrit;
    end
end
end

```

```

        hitsDxBBox = find(~logical(sum(hitsDxBBox')));
        p2=[p2(hitsDxBBox,1) p2(hitsDxBBox,2) p2(hitsDxBBox,3)\];
        newp = \[newp; p2\];
        Npoints = size(newp,1) ;
    else
        j=j+1;
    end
end
end

```

```

imf2=uint16(zeros(size(imf)));
for i =1:$NSitesInvVoi
    imf2(newp(i,1), newp(i,2), newp(i,3))=i;
end
clear p_othersx p_othersy p_othersz locs
clear CavSize A sz imf Npoints cond p p_curr
clear p_others removed i j sz Npoints hits xR yR zR
clear zRmin zRmax xRmin xRmax yRmin yRmax
clear zRmin2 zRmax2 xRmin2 xRmax2 yRmin2 yRmax2
clear d R LR hits hits2 xR2 yR2 zR2 LR2 R2
clear ans2 ans err1 p2 dxBBox hitsDxBBox newp p2
pack
MatlabExportList={'imf2'};
"

PairingMATLAB Apply snap 0 1
PairingMATLAB fire
remove PairingMATLAB

```

Grow Surface 6_26_11_nonDX.tcl

```

#Inputs: point or object to grow, extent to grow
#NR = number of points
#R = random points with pixel IDs
#RDist = total distance to radially grow point
#RNum_Dilate = number of times to dilate point at once

```

```

create HxUniformLabelField3
MS-Study setLabel RBS
set CubicImageSize [Thresh5p6 getDims]
set bbox [Thresh5p6 getBoundingBox]
RBS resize [lindex $CubicImageSize 0] [lindex $CubicImageSize 1] [lindex
$CubicImageSize 2]

```

```

RBS setBoundingBox [lindex $bbox 0] [lindex $bbox 1] [lindex $bbox 2] [lindex
$bbox 3] [lindex $bbox 4] [lindex $bbox 5]
create HxArithmetic RMask
RMask inputA connect RBS
RMask expr setValue "A<0"
RMask create
remove RMask
remove RBS
Result setLabel RBS

```

```

for {set k 1} {$k<= $NSitesInvVoi} {incr k} {

```

```

#Crop 5*RDist around random point to reduce comput.

```

```

#1. Dilate point/object
echo -> Diliating Random Point $k of $NSitesInvVoi...
#Label 0 as background; 1 as selected random point; 2 as the rest
create HxLabelVoxel RLabels
RLabels data connect R
RLabels regions setValue "s1 s2 s3"
RLabels boundary01 setValue [expr $k-1]
RLabels boundary12 setValue [expr $k]
RLabels create
remove RLabels

```

```

create HxGiEditor EDIT_isolate$k
R.Labels ImageData disconnect
EDIT_isolate$k attach R.Labels
EDIT_isolate$k selectMaterial2 2
EDIT_isolate$k subtract 2
EDIT_isolate$k selectMaterial2 1
EDIT_isolate$k add 1
EDIT_isolate$k detach
remove EDIT_isolate$k

```

```

R.Labels duplicate
R2.Labels setLabel seed

```

```

for {set j 1} {$j <= 2} {incr j} {

```

```

create HxLabelVoxel LV
LV data connect R.Labels
LV regions setValue "Background Signal"
LV boundary01 setValue 0
LV create
remove LV
R2.Labels ImageData disconnect
remove R.Labels
R2.Labels setLabel R.Labels

```

```

create HxGiEditor RDilate$k$j
RDilate$k$j attach R.Labels
RDilate$k$j selectMaterial2 1
for {set d 1} {$d < $RNum_Dilate} {incr d} {RDilate$k$j growSelection3D}
RDilate$k$j add 1
RDilate$k$j detach
remove RDilate$k$j
R.Labels duplicate
remove R.Labels
R2.Labels setLabel R.Labels

```

```

create HxCastField
CastField data connect R.Labels
CastField inputInfo send
CastField outputType setValue 0
CastField create
remove CastField R.Labels
R.to-byte setLabel R.Labels

```

```

set ntemp [[nonDxBSEThick5p6 duplicate] getLabel]
$ntemp setLabel BoneEdgeThicktemp
#2. Mask (1) with Thickened Bone Edge (BSEThick)
create HxArithmetic RMaskThick$j
RMaskThick$j inputA connect R.Labels
RMaskThick$j inputB connect BoneEdgeThicktemp
RMaskThick$j expr setValue "A.*B"
RMaskThick$j create
remove RMaskThick$j
Result setLabel RThickMask.Labels
remove BoneEdgeThicktemp

```

```

create HxArithmetic RMask$j

```

```

RMask$j inputA connect R.Labels
RMask$j inputB connect nonDxBSE5p6
RMask$j expr setValue "A.*B"
RMask$j create
remove RMask$j
remove R.Labels
Result setLabel R.Labels

```

```

#Convert R.Labels and seed to uint8 for transfer to matlab
create HxCastField
CastField data connect RThickMask.Labels
CastField inputInfo send
CastField outputType setValue 0
CastField create
remove CastField RThickMask.Labels
RThickMask.to-byte setLabel RThickMask.Labels

```

```

create HxCastField
CastField data connect seed
CastField inputInfo send
CastField outputType setValue 0
CastField create
remove CastField seed
seed.to-byte setLabel seed

```

```

while {[catch {[ create HxCalculus RanPMorphRecon ] getLabel} errmsg]} { create
HxCalculus RanPMorphRecon }
RanPMorphRecon A connect RThickMask.Labels
RanPMorphRecon A1 connect seed
RanPMorphRecon MatlabBuffer setValue "
Locs=find(A1);
sz = size(A1);
\[y,x,z]=ind2sub(sz, Locs);

A1(:,:,z(1))=imdilate(A1(:,:,z(1)), strel('disk', 5)).*A(:,:,z(1));
IM = imreconstruct(A1,A,26);
clear A A1 sz Locs x y z"
RanPMorphRecon Apply snap 0 1
RanPMorphRecon fire
remove RanPMorphRecon
if {$j == 1} { remove RThickMask.Labels }
IM setLabel RThickMorphRecon
RThickMorphRecon setBoundingBox [lindex $bbox 0] [lindex $bbox 1] [lindex $bbox
2] [lindex $bbox 3] [lindex $bbox 4] [lindex $bbox 5]

```

```

create HxArithmetic RMask2$k$j
RMask2$k$j inputA connect R.Labels
RMask2$k$j inputB connect R.ThickMorphRecon
RMask2$k$j expr setValue "A.*B"
RMask2$k$j create
remove RMask2$k$j
remove R.Labels
Result setLabel R.Labels
remove R.ThickMorphRecon

}

```

```

create HxLabelVoxel LABELS_$k$j
LABELS_$k$j data connect R.Labels
LABELS_$k$j regions setValue "Background Signal"
LABELS_$k$j boundary01 setValue 0
LABELS_$k$j create
remove LABELS_$k$j
R2.Labels ImageData disconnect
remove R.Labels
R2.Labels setLabel R.Labels

```

```

create HxLabelVoxel LABELSThick_$k$j
LABELSThick_$k$j data connect R.ThickMask.Labels
LABELSThick_$k$j regions setValue "Background Signal"
LABELSThick_$k$j boundary01 setValue 0
LABELSThick_$k$j create
remove LABELSThick_$k$j
R.ThickMask2.Labels ImageData disconnect
remove R.ThickMask.Labels
R.ThickMask2.Labels setLabel R.ThickMask.Labels

```

```

create HxArithmetic RBSadd
RBSadd inputA connect R.Labels

```



```

RBSadd inputB connect RBS
RBSadd expr setValue "$k*A+B"
RBSadd create
remove RBS
Result setLabel RBS
remove RBSadd

```

```

#Random point on BS dilated
#R.Labels setLabel RBS_$k
remove R.Labels seed RThickMask.Labels
}

```

```

Get Percent Damage Volume.tcl
#Amira script

```

```

#This script determines the percent damage volume DV/BV using 5.6 um thresholds and
2.8x2.5 um post test damage thresholds

```

```

set desktopDir "Q:/"
set scriptDir "$desktopDir\uDx/scripts/"
set dataDir "$desktopDir\uDx/DATA/"
set PTDPath "$desktopDir\uDx/$specimen/ROI/SPATIAL/PTDImages/"
set ThreshPath "$desktopDir\uDx/$specimen/ROI/Thresh"

```

```

#Load UV threshold Images at 0.7um
cd $ThreshPath
set imageNameThresh [glob "*Thresh*5p6.tif"]
set imageNameThresh [lindex $imageNameThresh 0]
set initIm "$ThreshPath/$imageNameThresh"
load -tif +box 0 1 0 1 0 1 +mode 2 $initIm
$imageNameThresh setLabel test
set imageSize5p6 [test getDims]
remove test
set filenamesThresh [glob "*Thresh*5p6*.tif"]
#set filenamesThresh [lrange $filenamesThresh 0 50]
set Nims [llength $filenamesThresh]
set name [[ eval load -tif +box 0 [expr 5.6*([lindex $imageSize5p6 0]-1)] 0 [expr
5.6*([lindex $imageSize5p6 1]-1)] 0 [expr 5*($Nims-1)] +mode 2 $filenamesThresh]
getLabel]
$name setLabel Thresh_$specimen

```

```

#Load post test damage (PTD) images and are 2.8um

```

```

#PTD images have been binary closed by ~70 um to group nearby damage pixels
cd $PTDPath
set imageNamePTD [glob "*PTD*.tif"]
set imageNamePTD [lindex $imageNamePTD 0]
set initIm "$PTDPath/$imageNamePTD"
load -tif +box 0 1 0 1 0 1 +mode 2 $initIm
$imageNamePTD setLabel test
set imageSize2p8 [test getDims]
remove test
set filenames [glob "*PTD*.tif"]
set filenamesPTD "$PTDPath/$filenames"
#set filenamesPTD [lrange $filenamesPTD 0 100]
set Nims [llength $filenamesPTD]
set name [lindex [[ eval load -tif +box 0 [expr 2.8*([lindex $imageSize2p8 0]-1)] 0 [expr
2.8*([lindex $imageSize2p8 1]-1)] 0 [expr [expr 2.5*($Nims-1)]] +mode 2
$filenamesPTD] getLabel] 0]
$name setLabel PTD_$specimen

```

```

#Dilate existing damage
#Label 0 as background; 1 as selected random point; 2 as the rest
create HxArithmetic
Arithmetic inputA connect Thresh_$specimen
Arithmetic expr setValue "A>0"
Arithmetic create
remove Arithmetic Thresh_$specimen
Result setLabel Thresh_$specimen

```

```

create HxArithmetic
Arithmetic inputA connect PTD_$specimen
Arithmetic expr setValue "A>0"
Arithmetic create
remove Arithmetic PTD_$specimen
Result setLabel PTD_$specimen

```

```

create HxGMC $curr\surfGen
$curr\surfGen data connect $curr
$curr\surfGen smoothing setValue 0
$curr\surfGen options setValue border 0
$curr\surfGen create
remove $curr\surfGen
$curr\surf setNoRename 1

```

```
set DV [PTD_$specimen\Labels getVolume 1]
set BV [Thresh_$specimen\Labels getVolume 1]
```

```
set DVBV [expr $DV/$BV]
```

```
create HxGiEditor PTDRestore
PTDRestore attach PTD_$specimen\Labels
gi selectMaterial2 0
gi growSelection3D
```

```
gi selectMaterial2 1
gi subtract 1
```

```
gi shrinkSelection3D
gi add 1
gi detach
remove gi
```

```
#Dilate until event is one cavity plus minimum association distance larger than original
for {set d 1} {$d <= 1} {incr d} {PTDRestore shrinkSelection3D}
PTDRestore add 1
PTDRestore detach
remove PTDRestore
```

Get Damage Site Shape Index 7_11_11.tcl

```
#Amira script
```

```
#This script determines the percent damage volume DV/BV using 5.6 um thresholds and
2.8x2.5 um post test damage thresholds
remove -all
```

```
set desktopDir "Q:/"
set scriptDir "$desktopDir\uDx/scripts/"
set dataDir "$desktopDir\uDx/DATA/"
set EPath "$desktopDir\uDx/$specimen/ROI/DV"
```

```
set imaxPrevious 0
for {set ROI 1} {$ROI <=4} {incr ROI} {
```

```

#Load post test damage (PTD) images and are 2.8um
#PTD images have been binary closed by ~70 um to group nearby damage pixels
cd $EPATH
set imageNameE [glob "*DamageEventsLabel*.tif"]
set imageNameE [lindex $imageNameE 0]
set initIm "$EPATH/$imageNameE"
load -tif +box 0 1 0 1 0 1 +mode 2 $initIm
$imageNameE setLabel test
set imageSize [test getDims]
remove test
set filenames [glob "*DamageEventsLabel*.tif"]
set filenamesE "$EPATH/$filenames"
#set filenamesPTD [lrange $filenamesPTD 0 100]
set filenamesE [lrange $filenamesE [expr 400*($ROI-1)-1] [expr 400*$ROI-1]-100]
set Nims [llength $filenamesE]
set name [lindex [[ eval load -tif +box 0 [expr 2.8*([lindex $imageSize 0]-1)] 0 [expr
2.8*([lindex $imageSize 1]-1)] 0 [expr [expr 2.5*($Nims-1)]] +mode 2 $filenamesE]
getLabel] 0]
$name setLabel D

```

```

#Dilate existing damage
#Label 0 as background; 1 as selected random point; 2 as the rest

set R [ D getRange]
set NSites [expr [lindex $R 1] - $imaxPrevious]

```

```

for {set i 1} {$i < $NSites} {incr i} {

```

```

#Label 0 as background; 1 as selected random point; 2 as the rest
create HxLabelVoxel DLabels
DLabels data connect D
DLabels regions setValue "s1 s2 s3"
DLabels boundary01 setValue [expr $i + $imaxPrevious ]
DLabels boundary12 setValue [expr $i +1 +$imaxPrevious]
DLabels create
remove DLabels

```

```

create HxGiEditor EDIT_$i
EDIT_$i attach D.Labels
EDIT_$i selectMaterial2 2
EDIT_$i subtract 2
EDIT_$i selectMaterial2 1

```

```
EDIT_$i add 1
EDIT_$i detach
remove EDIT_$i
```

```
#create HxGMC $curr\surfGen
#$curr\surfGen data connect D.Labels
#$curr\surfGen smoothing setValue 0
#$curr\surfGen options setValue border 1
#$curr\surfGen create
#remove $curr\surfGen
```

```
set DV [D.Labels getVolume 1]
set DS [D.Labels getSurfaceArea 1]
```

```
if { $DS <= 0 } { } else {
set Aspect [expr $DS/$DV]
```

```
set outfile [open "$dataDir$specimen\_DamageSiteDetails.txt" a]
puts $outfile "$i\t\t\t$DV\t\t\t$DS\t\t\t$Aspect"
close $outfile
```

```
remove D.Labels
}
}
remove D
set imaxPrevious $NSites
}
```

Appendix 1.4 Resorption Cavity Morphology Quantification

Cavity Depth.tcl

```
#Amira script
```

```
#Measure cavity depth in HUMAN SPECIMENS
#Develop CJ Slyfield
#6-13-2011
```

```
#In no way whatsoever was Evgeniy Vladimir Tkachenko involved in the development of
this code
```

```
remove -all
```

```
set desktopDir "Q:/"
```

```

set scriptDir "$desktopDir\uDx/scripts/"
set funcDir "$desktopDir\uDx/scripts/functions/"
set SOIDir "$desktopDir\uDx/$specimen\ROI/RESORPTION/"
set cavDir "$desktopDir\uDx/$specimen\ROI/RESORPTION/Cavities/"
set GlobalDir "$desktopDir\uDx/$specimen\ROI/RESORPTION/Cavities/"
set dataDir "$desktopDir\uDx/DATA/"
file mkdir "$cavDir/Depth"

```

```

#dist = min dist (um) to bounding box
set dist 5
#Number of times to dilate the cavity boundary at once
set CavNDilate 15

```

#GET CAVITY DEPTH FOR EACH SLICE OF INTEREST

```

#Find cavity files for current slice of interest
cd $GlobalDir
set filenames [glob "*SOI$SOI\*GlobalCoords*"]
set NCavs [llength $filenames]
#FOR ALL CAVITIES IN SLICE OF INTEREST$SOI

for {set i $imin} {$i <= $NCavs} {incr i} {

    set TclLoadName [glob "SOI_$SOI*.tcl"]
    #Determine slices loaded for labeling current cavity
    source $cavDir[index $TclLoadName [expr $i-1]]

    #Load cavity file
    cd $GlobalDir
    set CavFileNames [glob "$specimen\_TraceCav*SOI$SOI\_CavityID\.am"]
    set CurrentCavity [[[load $CavFileNames] setLabel
Cav_Orig_$specimen\_CavityID] getLabel]
    set CavBBox [Cav_Orig_$specimen\_CavityID getBoundingBox]

```

```

#Load Bone Thresh (every cavity requires different numbers of slices above and below
SOI)
if { [expr $Grown == 0] } {
    [load
"$SOI_dir\Cv_Ar_$SOI\$_specimen\_SOI_$SOI\_Cv_$CavityID\_Thresh.am"] setLabel
Thresh
    } else {
        [load "$SOI_dir\Cv_Ar_$SOI\$_specimen\_SOI_$SOI\_Cv_$CavityID\_Thresh.am"] setLabel Thresh
    }
}

```

```

set CavOrigAreaCurr [Cav_Orig_$_specimen\_$_CavityID getArea]

```

```

#If the cavity is too broad, it will take ~20Gb of ram in matlab and it will generate depth
errors later

```

```

if { $CavOrigAreaCurr > 320000 } {

```

```

#Remove all remaining variables before next loop iteration
remove -all
set outfile [open "$dataDir$_specimen\_GetAreasOnly.txt" a]
puts $outfile "$SOI\t$_CavityID"
close $outfile

```

```

} else {

```

```

    #Cavity surface should already have the global positioning coordinates
    #Cav_Orig_$_specimen\_$_i setBoundingBox $CavXmin $CavXmax $CavYmin
    $CavYmax $CavZmin $CavZmax

```

```

    #Use save .am bone thresh to find the edge image
    while { [catch { [ create HxCalculus Edger ] getLabel } errmsg] } { create HxCalculus
    Edger }
    Edger A connect Thresh
    Edger MatlabBuffer setValue "
    s=size(A);
    imf=false(s(1),s(2),s(3));

```

```

for ii=1:s(3)
    %Updated 6 20 11 to smooth the outer boundary of the bone before edging
    imf(:, :, ii)=edge(imclose(A(:, :, ii), strel('disk', 5)), 'canny');
end
imf=uint8(imf);
MatlabExportList = {'imf'};
"

Edger Apply snap 0 1
Edger fire
remove Edger
imf setLabel $specimen\_Edge
set bboxOrig [Thresh getBoundingBox]
set xmin [lindex $bboxOrig 0]
set ymin [lindex $bboxOrig 2]
set zmin [lindex $bboxOrig 4]
set xmax [lindex $bboxOrig 1]
set ymax [lindex $bboxOrig 3]
set zmax [lindex $bboxOrig 5]
$specimen\_Edge setBoundingBox $xmin $xmax $ymin $ymax $zmin $zmax

remove CavityTracingError CavityTracingError2

set NPointsCurr [Cav_Orig_$specimen\_CavityID getNumPoints]

create HxLineSet
LineSet setLabel locs
for {set p 1} {$p < $NPointsCurr} {incr p} {
    set vert [Cav_Orig_$specimen\_CavityID getPoint $p]
    locs addPoint [lindex $vert 0] [lindex $vert 1] [lindex $vert 2]
}

#Generate voxel image of cavity and get cavity boundary
while {[catch {[ create HxCalculus CreateVoxellImageFromCav ] getLabel} errmsg]} {
    create HxCalculus CreateVoxellImageFromCav }
CreateVoxellImageFromCav A connect Cav_Orig_$specimen\_CavityID
CreateVoxellImageFromCav A1 connect locs
CreateVoxellImageFromCav MatlabBuffer setValue "
p=(lineset.points');
x=round(p(:,1)/.7);
y=round(p(:,2)/.7);

```



```

z=round(p(:,3)/5);

xmin=min(x);
ymin=min(y);
zmin=min(z);
xmax=max(x);
ymax=max(y);
zmax=max(z);

%im=uint16(zeros(ceil(ymax-ymin), ceil(xmax-xmin), ceil(zmax-zmin)));
im=uint8(zeros(ceil(xmax-xmin),ceil(ymax-ymin), ceil(zmax-zmin)));
im2=uint8(zeros(ceil(xmax-xmin)+round(2*$CavNDilate/.7), ceil(ymax-
ymin)+round(2*$CavNDilate/.7), ceil(zmax-zmin)+round(2*$CavNDilate/5)));

for i = 1:length(x)
    %im(y(i)-ymin+1, x(i)-xmin+1, z(i)-zmin+1) = 1;
    im(x(i)-xmin+1, y(i)-ymin+1, z(i)-zmin+1) = 1;
    im2(x(i)-xmin+round($CavNDilate/.7), y(i)-ymin+round($CavNDilate/.7), z(i)-
zmin+round($CavNDilate/5)) = 1;
end
sz=size(im);

clear x y z xmin ymin zmin xmax ymax zmax p
MatlabExportList = {'im' , 'im2'};
"

CreateVoxelImageFromCav Apply snap 0 1
CreateVoxelImageFromCav fire
remove CreateVoxelImageFromCav
im setLabel VoxelCav_$CavityID
im2 setLabel VoxelCav_$CavityID\ForDilation

set size [VoxelCav_$CavityID getDims]
set bboxOrig [Cav_Orig_$specimen\_ $CavityID getBoundingBox]
set sizeDil [VoxelCav_$CavityID\ForDilation getDims]
set xOrig [lindex $bboxOrig 0]
set yOrig [lindex $bboxOrig 2]
set zOrig [lindex $bboxOrig 4]
VoxelCav_$CavityID setBoundingBox $xOrig [expr $xOrig+0.7*([lindex $size 0]-1)]
$yOrig [expr $yOrig+0.7*([lindex $size 1]-1)] $zOrig [expr $zOrig+5*([lindex $size 2]-
1)]

set xmin [expr [lindex $bboxOrig 0] - [expr $CavNDilate]]
set ymin [expr [lindex $bboxOrig 2]- [expr $CavNDilate]]
set zmin [expr [lindex $bboxOrig 4]- [expr $CavNDilate]]
set xmax [expr [lindex $bboxOrig 1] + [expr $CavNDilate]]
set ymax [expr [lindex $bboxOrig 3] + [expr $CavNDilate]]

```

```

set zmax [expr [lindex $bboxOrig 5] + [expr $CavNDilate]]
VoxelCav_${CavityID}\ForDilation setBoundingBox $xmin $xmax $ymin $ymax $zmin
$zmax

```

```

set GrowName [[ $specimen\_Edge duplicate] getLabel]
$GrowName setLabel GrowCav${CavityID}\_voxel
set GrowCavBBox [VoxelCav_${CavityID}\ForDilation getBoundingBox]
set transform [Thresh getBoundingBox]
set size [Thresh getDims]
set xmin [expr [lindex $GrowCavBBox 0]/0.7 - [lindex $transform 0]/0.7]
set xmax [expr [lindex $GrowCavBBox 1]/0.7 - [lindex $transform 0]/0.7]
set ymin [expr [lindex $GrowCavBBox 2]/0.7 - [lindex $transform 2]/0.7]
set ymax [expr [lindex $GrowCavBBox 3]/0.7 - [lindex $transform 2]/0.7]

```

#UPDATED CROP 6-25

```

#wASSSSSSSSS GrowCav${i}\_voxel crop $xmin $ymin $xmax $ymax 0 [lindex $size
2]
GrowCav${CavityID}\_voxel crop $xmin $xmax $ymin $ymax 0 [lindex $size 2]

```

```

create HxArithmetic
Arithmetic inputA connect GrowCav${CavityID}\_voxel
Arithmetic expr setValue "A>0"
Arithmetic create
remove Arithmetic
remove GrowCav${CavityID}\_voxel

```

```

Result setLabel GrowCav$CavityID\_BoneEdge

#Resample to cubic voxels
create HxResample
Resample data connect VoxelCav_$CavityID\.ForDilation
Resample data send
Resample mode setValue 0
set matchSize [GrowCav$CavityID\_BoneEdge getDims]
Resample resolution setValues [lindex $matchSize 0] [lindex $matchSize 1] [lindex
$matchSize 2]
Resample filter setValue 0
Resample create
remove Resample
remove VoxelCav_$CavityID\.ForDilation
VoxelCav_$CavityID\.Resampled setLabel VoxelCav_$CavityID\.ForDilation

create HxResample
Resample data connect GrowCav$CavityID\_BoneEdge
Resample data send
Resample mode setValue 1
Resample voxelSize setValues 1 1 1
Resample filter setValue 0
Resample create
remove Resample
remove GrowCav$CavityID\_BoneEdge
GrowCav$CavityID\_BoneEdge.Resampled setLabel GrowCav$CavityID\_BoneEdge

create HxResample
Resample data connect VoxelCav_$CavityID\.ForDilation
Resample data send
Resample mode setValue 1
Resample voxelSize setValues 1 1 1
Resample filter setValue 0
Resample create
remove Resample
remove VoxelCav_$CavityID\.ForDilation
VoxelCav_$CavityID\.Resampled setLabel VoxelCav_$CavityID\.ForDilation

source "$funcDir/CavDepthGrowSurface_06_19_11.tcl"

#Create surface of interpolated bone surface
create HxGMC CavSurfGen$CavityID
CavSurfGen$CavityID data connect Cav$CavityID\_Outer
CavSurfGen$CavityID smoothing setValue 0
CavSurfGen$CavityID options setValue border 1
CavSurfGen$CavityID create

```

remove CavSurfGen\$CavityID

#Determine the main surface normal direction of cavity
#simplify the surfave to remove surface staircase effect
set CavOuterDupName [[Cav\$CavityID_Outer.surf duplicate] getLabel]
\$CavOuterDupName setLabel Cav\$CavityID_Outer_LowRes

create HxSimplifier
Simplifier attach Cav\$CavityID_Outer_LowRes
Simplifier simplifyParameters setValue 0 [expr 0.1*[Cav\$CavityID_Outer.surf
getNumTriangles]]
Simplifier simplifyOptions setValue 0 1
Simplifier fire
Simplifier simplifyAction setValue 0
Simplifier simplifyAction send
Simplifier detach

set CavSimpName [[Cav_Orig_\$specimen_ \$CavityID duplicate] getLabel]
\$CavSimpName setLabel Cav\$CavityID_LowRes
create HxSimplifier
Simplifier attach Cav\$CavityID_LowRes
Simplifier simplifyParameters setValue 0 [expr 0.1*[Cav_Orig_ \$specimen_ \$CavityID
getNumTriangles]]
Simplifier simplifyOptions setValue 0 1
Simplifier fire
Simplifier simplifyAction setValue 0
Simplifier simplifyAction send
Simplifier detach

#calc surf normals
create HxSurfaceNormals {SurfaceNormals}
SurfaceNormals Surface connect Cav\$CavityID_LowRes
SurfaceNormals fire
SurfaceNormals NormalType setValue 0
SurfaceNormals InvertedNormal setValue 0 0
SurfaceNormals InvertedNormal setToggleVisible 0 1
SurfaceNormals CreaseAngle setMinMax -3.14159274101257 3.14159274101257
SurfaceNormals CreaseAngle setButtons 0
SurfaceNormals CreaseAngle setIncrement 0.418879
SurfaceNormals CreaseAngle setValue 0
SurfaceNormals CreaseAngle setSubMinMax -3.14159274101257 3.14159274101257
SurfaceNormals doIt snap 0 1
SurfaceNormals fire
remove SurfaceNormals
Cav\$CavityID_LowRes.vertex-normals setLabel vectors

```

source "$funcDir/Iteratively_Pointwrap_6_19_2011.tcl"

if {[expr [Cav$CavityID\_Outer\_PW getNumPoints]==0]} {} else {

source "$funcDir/Iteratively_Simplify_6_19_2011.tcl"


create HxLineSet
LineSet setLabel d


for {set j 1} {$j < [Cav$CavityID\_Outer\_PW\_Simple getNumPoints]} {incr j} {
    set q [vectors getValue $j]
    d addPoint [lindex $q 0] [lindex $q 1] [lindex $q 2]
}


set NInterpUpsample 1

source "$funcDir/CavDepthInterpolateSurface_CJS_07_10_11.tcl"

if {[catch {[ InterpCavVox_$CavityID getLabel]} errmsg]} {

} else {


source "$funcDir/CavDepth_VolandErrors_CJS_6_19_11.tcl"
set xmin0 $xmin
set ymin0 $ymin
set zmin0 $zmin
set xmax0 $xmax
set ymax0 $ymax
set zmax0 $zmax
CavVol$CavityID setBoundingBox $xmin0 $xmax0 $ymin0 $ymax0 $zmin0 $zmax0

```

```
#points2add setBoundingBox $xmin0 $xmax0 $ymin0 $ymax0 $zmin0 $zmax0
points2add setBoundingBox $xmin $xmax $ymin $ymax $zmin $zmax
```

```
#Get Cavity Volume --- must now close the cav vol image with strel of radius = max
Cv.De
#to ensure the entire cavity is filled
remove SurfaceDistance distance points2add.surf displacements adjust
remove CavVol$CavityID CavVol$CavityID\surf
```

```
#calc surf normals
create HxSurfaceNormals {SurfaceNormals}
SurfaceNormals Surface connect Cav$CavityID\_LowRes
SurfaceNormals fire
SurfaceNormals NormalType setValue 0
SurfaceNormals InvertedNormal setValue 0 0
SurfaceNormals InvertedNormal setToggleVisible 0 1
SurfaceNormals CreaseAngle setMinMax -3.14159274101257 3.14159274101257
SurfaceNormals CreaseAngle setButtons 0
SurfaceNormals CreaseAngle setIncrement 0.418879
SurfaceNormals CreaseAngle setValue 0
SurfaceNormals CreaseAngle setSubMinMax -3.14159274101257 3.14159274101257
SurfaceNormals doIt snap 0 1
SurfaceNormals fire
remove SurfaceNormals
Cav$CavityID\_LowRes.vertex-normals setLabel vectors
```

```
create HxLineSet
LineSet setLabel d
for {set j 1} {$j < [Cav$CavityID\_Outer_PW_Simple getNumPoints]} {incr j} {
    set q [vectors getValue $j]
    d addPoint [lindex $q 0] [lindex $q 1] [lindex $q 2]
}
}
```

```
source "$funcDir\CavDepth_Vol_CJS_6_19_11.tcl"
CavVol$CavityID setBoundingBox $xmin0 $xmax0 $ymin0 $ymax0 $zmin0 $zmax0
```

```
set CavTraceError 0
set AlgorithmicErr 0
```

```
#Determine max cavity depth to know how much to close CavVol
create HxGMC CavSurfGen$CavityID
CavSurfGen$CavityID data connect CavVol$CavityID
CavSurfGen$CavityID smoothing setValue 0
CavSurfGen$CavityID options setValue border 1
CavSurfGen$CavityID create
remove CavSurfGen$CavityID
```

```
create HxSimplifier
Simplifier attach CavVol$CavityID\surf
Simplifier simplifyParameters setValue 0 80000
Simplifier simplifyOptions setValue 0 1
Simplifier fire
Simplifier simplifyAction setValue 0
Simplifier simplifyAction send
Simplifier detach
```

```
create HxGMC CavSurfGen$CavityID
CavSurfGen$CavityID data connect InterpCavVox_$CavityID
CavSurfGen$CavityID smoothing setValue 0
CavSurfGen$CavityID options setValue border 1
CavSurfGen$CavityID create
remove CavSurfGen$CavityID
```

```
set InterpDupName [[InterpCavVox_$CavityID\surf duplicate] getLabel]
$InterpDupName setLabel surf
source "$funcDir\Interp_Pointwrap_6_19_2011.tcl"
remove InterpCavVox_$CavityID\surf
surf_PW setLabel InterpCavVox_$CavityID\surf
```

```
create HxSimplifier
Simplifier attach InterpCavVox_$CavityID\surf
Simplifier simplifyParameters setValue 0 80000
Simplifier simplifyOptions setValue 0 1
Simplifier fire
```

```
Simplifier simplifyAction setValue 0
Simplifier simplifyAction send
Simplifier detach
```

```
create HxSurfaceDistance
SurfaceDistance surface1 connect CavVol$CavityID\surf
SurfaceDistance surface2 connect InterpCavVox_$CavityID\surf
SurfaceDistance output setValue 1 1
SurfaceDistance output setValue 0 1
SurfaceDistance action snap
SurfaceDistance action send
set maxDistance [SurfaceDistance getMax]
```

```
if {[expr $maxDistance>300]} {set maxDistance 100}
```

```
create HxLabelVoxel CavLabels
CavLabels data connect CavVol$CavityID
CavLabels regions setValue "Background Signal"
CavLabels boundary01 setValue 0
CavLabels create
remove CavLabels
create HxGiEditor CavVolClose
CavVolClose attach CavVol$CavityID\Labels
CavVolClose selectMaterial2 1
for {set d 1} {$d < $maxDistance} {incr d} {CavVolClose growSelection3D}
for {set d 1} {$d < $maxDistance} {incr d} {CavVolClose shrinkSelection3D}
CavVolClose add 1
CavVolClose detach
remove CavVolClose
```

```
distance setLabel CvDepth$CavityID
displacements setLabel CvVecs$CavityID
#source "Q:/Cavity Depth/scripts/CavityDepthGetMaxDepth_CJS_6_10_11.tcl"
set CavDepthRMSCurr [SurfaceDistance getMean]
set CavDepthMeanCurr [SurfaceDistance getMean]
set CavDepthMaxCurr [SurfaceDistance getMax]
lappend CavDepthRMS [SurfaceDistance getMean]
```



```

lappend CavDepthMean [SurfaceDistance getMean]

remove VertexCvDeMa Dmax VertexCvDeMa2 Dmax2 vex locs i maindir depth

CavVol$CavityID\Labels ImageData disconnect

create HxCastField
CastField data connect CavVol$CavityID\Labels
CastField outputType setValue 0
CastField create
remove CastField CavVol$CavityID\Labels
CavVol$CavityID\to-byte setLabel CavVol$CavityID\Labels

#Create surface to get volume of cavity
create HxGMC CavSurfGen$CavityID
CavSurfGen$CavityID data connect CavVol$CavityID\Labels
CavSurfGen$CavityID smoothing setValue 2
CavSurfGen$CavityID options setValue border 1
CavSurfGen$CavityID create
remove CavSurfGen$CavityID

set CavVolCurr [CavVol$CavityID\surf getVolume]
lappend CavVolumes $CavVolCurr
Cav_Orig_$specimen\_CavityID setNoRename 1
Cav_Orig_$specimen\_CavityID removeUnusedPoints
Cav_Orig_$specimen\_CavityID save "HxSurface binary"
"$GlobalDir\Depth/$specimen\_Cav_Orig_SOI_$SOI\_CavityID.am"

CavVol$CavityID setNoRename 1
CavVol$CavityID save "AmiraMesh ZIP"
"$GlobalDir\Depth/$specimen\_Cv.V_$SOI\_CavityID.am"

Cav$CavityID\_Outer_PW_Simple setNoRename 1
Cav$CavityID\_Outer_PW_Simple removeUnusedPoints
Cav$CavityID\_Outer_PW_Simple save "HxSurface binary"
"$GlobalDir\Depth/$specimen\_Cv.De_Points_SOI_$SOI\_CavityID.am"

```

```

if {[catch {[InterpCavVox_$CavityID\.surf removeUnusedPoints] } errmsg ]} {}
if {[catch {[InterpCavVox_$CavityID\.surf setNoRename 1] } errmsg ]} {}
if {[catch {[InterpCavVox_$CavityID\.surf save "HxSurface binary"
"$GlobalDir\Depth/$specimen\_CvInterpSurf_SOI_$SOI_$CavityID\.am"] } errmsg]}
{
#Create surface to get volume of cavity
create HxGMC CavSurfGen$CavityID
CavSurfGen$CavityID data connect InterpCavVox_$CavityID
CavSurfGen$CavityID smoothing setValue 0
CavSurfGen$CavityID options setValue border 1
CavSurfGen$CavityID create
remove CavSurfGen$CavityID
InterpCavVox_$CavityID\.surf setNoRename 1
InterpCavVox_$CavityID\.surf removeUnusedPoints
InterpCavVox_$CavityID\.surf save "HxSurface binary"
"$GlobalDir\Depth/$specimen\_CvInterpSurf_SOI_$SOI_$CavityID\.am" }

#Remove all remaining variables before next loop iteration
remove -all
set outfile [open "$dataDir$specimen\_CavityData.txt" a]
puts $outfile
"$SOI\t\t$CavityID\t\t\t$CavDepthMaxCurr\t\t\t$CavOrigAreaCurr\t\t\t$CavVolCurr\t\t\t"
"
close $outfile

}
#End of pw check loop
remove -all
}

}
#End of i = 1:N cavs loop

remove -all

}
#End of for SOI loop

```

CavityDepthGrowSurface.tcl

#Inputs: point or object to grow, extent to grow

```

#NR = number of points
#R = random points with pixel IDs
#RDist = total distance to radially grow point
#RNum_Dilate = number of times to dilate point at once
set dilate [expr $CavNDilate /2]

#1. Dilate cavity along bone surface
create HxLabelVoxel CavLabels
CavLabels data connect VoxelCav_$CavityID\ForDilation
CavLabels regions setValue "Background Signal"
CavLabels boundary01 setValue 0
CavLabels create
remove CavLabels

create HxGiEditor EDIT$CavityID\_Dilate1
EDIT$CavityID\_Dilate1 attach VoxelCav_$CavityID\Labels
EDIT$CavityID\_Dilate1 selectMaterial2 1
for {set d 1} {$d < [expr $dilate+4]} {incr d} {EDIT$CavityID\_Dilate1
growSelection3D}
EDIT$CavityID\_Dilate1 add 1
EDIT$CavityID\_Dilate1 detach
remove EDIT$CavityID\_Dilate1

#2. Mask (1) with Bone Edge
create HxArithmetic BSMask
BSMask inputA connect GrowCav$CavityID\_BoneEdge
BSMask inputB connect VoxelCav_$CavityID\Labels
BSMask expr setValue "A.*B"
BSMask create
remove BSMask

Result setLabel BS_dilate_$CavityID\_pass1
set Pass1Name [[BS_dilate_$CavityID\_pass1 duplicate] getLabel]
$Pass1Name setLabel FirstPassDelete

remove VoxelCav_$CavityID\Labels

create HxLabelVoxel CavLabels
CavLabels data connect BS_dilate_$CavityID\_pass1
CavLabels regions setValue "Background Signal"
CavLabels boundary01 setValue 0
CavLabels create
remove CavLabels

create HxGiEditor EDIT$CavityID\_Dilate2

```

```

EDIT$CavityID\_Dilate2 attach BS_dilate_$CavityID\_pass1.Labels
EDIT$CavityID\_Dilate2 selectMaterial2 1
for {set d 1} {$d < $dilate} {incr d} {EDIT$CavityID\_Dilate2 growSelection3D}
EDIT$CavityID\_Dilate2 add 1
EDIT$CavityID\_Dilate2 detach
remove EDIT$CavityID\_Dilate2

```

```

#2. Mask (1) with Bone Edge
create HxArithmetic BSMask
BSMask inputA connect GrowCav$CavityID\_BoneEdge
BSMask inputB connect BS_dilate_$CavityID\_pass1.Labels
BSMask expr setValue "A.*B"
BSMask create
remove BSMask

```

```

Result setLabel BS_dilate_$CavityID\_pass2
remove BS_dilate_$CavityID\_pass1.Labels
remove BS_dilate_$CavityID\_pass1

```

```

create HxLabelVoxel CavLabels
CavLabels data connect BS_dilate_$CavityID\_pass2
CavLabels regions setValue "Background Signal"
CavLabels boundary01 setValue 0
CavLabels create
remove CavLabels

```

```

create HxGiEditor EDIT$CavityID\_Dilate3
EDIT$CavityID\_Dilate3 attach BS_dilate_$CavityID\_pass2.Labels
EDIT$CavityID\_Dilate3 selectMaterial2 1
for {set d 1} {$d < $dilate} {incr d} {EDIT$CavityID\_Dilate3 growSelection3D}
EDIT$CavityID\_Dilate3 add 1
EDIT$CavityID\_Dilate3 detach
remove EDIT$CavityID\_Dilate3

```

```

#2. Mask (1) with Bone Edge
create HxArithmetic BSMask
BSMask inputA connect GrowCav$CavityID\_BoneEdge
BSMask inputB connect BS_dilate_$CavityID\_pass2.Labels
BSMask inputC connect FirstPassDelete
BSMask expr setValue "(A.*B)-C"
BSMask create
remove BSMask

```

```
Result setLabel BS_dilate_${CavityID}\_pass3
remove BS_dilate_${CavityID}\_pass2.Labels
remove VoxelCav_${CavityID}.Labels
remove BS_dilate_${CavityID}\_pass2
```

```
create HxLabelVoxel CavLabels
CavLabels data connect BS_dilate_${CavityID}\_pass3
CavLabels regions setValue "Background Signal"
CavLabels boundary01 setValue 0
CavLabels create
remove CavLabels
```

```
create HxGiEditor EDIT${CavityID}\_Dilate4
EDIT${CavityID}\_Dilate4 attach BS_dilate_${CavityID}\_pass3.Labels
EDIT${CavityID}\_Dilate4 selectMaterial2 1
for {set d 1} {$d < $dilate} {incr d} {EDIT${CavityID}\_Dilate4 growSelection3D}
EDIT${CavityID}\_Dilate4 add 1
EDIT${CavityID}\_Dilate4 detach
remove EDIT${CavityID}\_Dilate4
```

```
#2. Mask (1) with Bone Edge
create HxArithmetic BSMask
BSMask inputA connect GrowCav${CavityID}\_BoneEdge
BSMask inputB connect BS_dilate_${CavityID}\_pass3.Labels
BSMask expr setValue "A.*B"
BSMask create
remove BSMask
```

```
Result setLabel BS_dilate_${CavityID}\_pass4
remove BS_dilate_${CavityID}\_pass3.Labels
remove BS_dilate_${CavityID}\_pass3
```

```
#1. Dilate cavity along bone surface
create HxLabelVoxel CavLabels
CavLabels data connect VoxelCav_${CavityID}.ForDilation
CavLabels regions setValue "Background Signal"
CavLabels boundary01 setValue 0
CavLabels create
remove CavLabels
```

```
create HxGiEditor EDIT${CavityID}\_Dilate1
EDIT${CavityID}\_Dilate1 attach VoxelCav_${CavityID}.Labels
EDIT${CavityID}\_Dilate1 selectMaterial2 1
for {set d 1} {$d < 4} {incr d} {EDIT${CavityID}\_Dilate1 growSelection3D}
```

```

EDIT$CavityID\_Dilate1 add 1
EDIT$CavityID\_Dilate1 detach
remove EDIT$CavityID\_Dilate1

```

```

create HxArithmetic CavMask
CavMask inputA connect BS_dilate_$CavityID\_pass4
CavMask inputB connect VoxelCav_$CavityID\.Labels
CavMask expr setValue "A-B"
CavMask create
remove CavMask

```

```

Result setLabel Cav$CavityID\_Outer
remove VoxelCav_$CavityID\.ForDilation BS_dilate_$CavityID\_pass4
remove FirstPassDelete

```

IterativelySimplify.tcl

#This script iteratively simplifies surfaces until they contain at least 1000 vetices

```

create HxSimplifier
set PwName [[Cav$CavityID\_Outer\_PW duplicate] getLabel]
$PwName setLabel Cav$CavityID\_Outer\_PW\_temp
Simplifier attach Cav$CavityID\_Outer\_PW\_temp
Simplifier simplifyOptions setValue 0 1
Simplifier simplifyParameters setValue 0 650
set CurrSimp [Simplifier simplifyParameters getValue]
Simplifier fire
Simplifier simplifyAction setValue 0
Simplifier simplifyAction send
Simplifier detach

while {[expr [Cav$CavityID\_Outer\_PW\_temp getNumPoints] < 1000 ]} {

    remove Cav$CavityID\_Outer\_PW\_temp
    create HxSimplifier
    set PwName [[ Cav$CavityID\_Outer\_PW duplicate] getLabel]
    $PwName setLabel Cav$CavityID\_Outer\_PW\_temp
    Simplifier attach Cav$CavityID\_Outer\_PW\_temp
    Simplifier simplifyOptions setValue 0 1
    Simplifier simplifyParameters setValue 0 [expr 1.15*$CurrSimp]
    set CurrSimp [Simplifier simplifyParameters getValue]
    Simplifier fire
}

```

```

Simplifier simplifyAction setValue 0
Simplifier simplifyAction send
Simplifier detach

}

remove Cav$CavityID\_Outer\_PW
Cav$CavityID\_Outer\_PW\_temp setLabel Cav$CavityID\_Outer\_PW\_Simple

CavDepthInterpolateSurface.tcl

#Create Surface interpolation across cavity in matlab
while {[catch {[ create HxCalculus InterpolateCavity] getLabel} errmsg]} {
    create HxCalculus InterpolateCavity }
InterpolateCavity A connect Cav$CavityID\_Outer\_PW\_Simple
InterpolateCavity A1 connect d
#PairingMATLAB A1 connect dxBBox
InterpolateCavity MatlabBuffer setValue "
\[v,p\]=max(abs(mean(lineset.points')));
mainDirection = p;
m=(mean(lineset.points'));
mainDirectionSigned =sign(m(p));

warning off
g=double(surf.vertices);
if mainDirection == 3
    gx=g(:,1);
    gy=g(:,2);
    gz=g(:,3);
elseif mainDirection == 2
    gx=g(:,1);
    gy=g(:,3);
    gz=g(:,2);
else
    gx=g(:,3);
    gy=g(:,2);
    gz=g(:,1);
end

ny = $NInterpUpsample*round(max(gy)-min(gy));
nx = $NInterpUpsample*round(max(gx)-min(gx));
\[x,y\]=meshgrid(linspace(min(gx), max(gx),nx),linspace(min(gy),max(gy),ny));

```

```

\st,p\]=tpaps ( \gx'; gy^\], gz');
%st=tpaps ( \gx'; gy^\], gz', 1/5*p);

h=fnval(st, \[x(:), y(:)\]');
X=round($NInterpUpsample*x(:));
Y=round($NInterpUpsample*y(:));

Xmin= floor($NInterpUpsample*min(x(:)));
Ymin= floor($NInterpUpsample*min(y(:)));
Zmin=ceil(min(h));

if mainDirection ==3
    InterpolatedCav=uint8(zeros(max(X)-Xmin+1, max(Y)-
Ymin+1,ceil($NInterpUpsample*(max(h)-min(h)))));
    for i =1:length(x(:))

        if round($NInterpUpsample*( h(i)-Zmin+1)) < 1
            InterpolatedCav(X(i)-Xmin+1, Y(i)-Ymin+1, 1)=1;
        elseif round($NInterpUpsample*( h(i)-Zmin+1)) >
ceil($NInterpUpsample*(max(h)-min(h)))
            InterpolatedCav(X(i)-Xmin+1, Y(i)-Ymin+1, ceil($NInterpUpsample*(max(h)-
min(h)))) =1;
        else
            InterpolatedCav(X(i)-Xmin+1, Y(i)-Ymin+1, round($NInterpUpsample*( h(i)-
Zmin+1)))=1;
        end
    end

    %The xMin above is rounded
    mins=\[min(x(:)) min(y(:)) min(h)\];
    maxs =\[ max(x(:)) max(y(:)) max(h) \];

elseif mainDirection == 2
    InterpolatedCav=uint8(zeros(max(X)-Xmin+1, ceil($NInterpUpsample*(max(h)-
min(h))), max(Y)-Ymin+1));
    for i =1:length(x(:))
        if round($NInterpUpsample*( h(i)-Zmin+1)) < 1
            InterpolatedCav(X(i)-Xmin+1, 1 ,Y(i)-Ymin+1)=1;
        elseif round($NInterpUpsample*( h(i)-Zmin+1)) >
ceil($NInterpUpsample*(max(h)-min(h)))
            InterpolatedCav(X(i)-Xmin+1, ceil($NInterpUpsample*(max(h)-min(h))), Y(i)-
Ymin+1)=1;
        else

```



```

        InterpolatedCav(X(i)-Xmin+1, round($NInterpUpsample*( h(i)-Zmin+1)), Y(i)-
Ymin+1)=1;
    end
end
mins=\[min(x(:)) min(h) min(y(:))\];
maxs =\[ max(x(:)) max(h) max(y(:)) \];

else
    InterpolatedCav=uint8( zeros(ceil($NInterpUpsample*(max(h)-min(h))), max(Y)-
Ymin+1,max(X)-Xmin+1));

for i =1:length(x(:))
    if round($NInterpUpsample*( h(i)-Zmin+1)) < 1
        InterpolatedCav( 1, Y(i)-Ymin+1,X(i)-Xmin+1 )=1;
    elseif round($NInterpUpsample*( h(i)-Zmin+1)) > ceil($NInterpUpsample*(max(h)-
min(h)))
        InterpolatedCav(ceil($NInterpUpsample*(max(h)-min(h))), Y(i)-Ymin+1,X(i)-
Xmin+1 )=1;
    else
        InterpolatedCav(round($NInterpUpsample*( h(i)-Zmin+1)) ,Y(i)-Ymin+1 ,X(i)-
Xmin+1 )=1;
    end
end
end

mins=\[ min(h) min(y(:)) min(x(:)) \];
maxs =\[ max(h) max(y(:)) max(x(:)) \];
end

MatlabExportList = {'InterpolatedCav', 'mins','mainDirection', 'mainDirectionSigned'};
clear gof cfun output v p
"
InterpolateCavity Apply snap 0 1
InterpolateCavity fire
remove InterpolateCavity

```

```

InterpolatedCav setLabel InterpCavVox_ $CavityID

```

```

#Duplicate and use for cavity depth only (wont work for Cv.V)
set IntCavDupName [[InterpCavVox_ $CavityID duplicate ] getLabel]
$IntCavDupName setLabel InterpCavVox_ $CavityID\_Wide

set OuterBBox [Cav$CavityID\_Outer.surf getBoundingBox]
set InterpCavSize [InterpCavVox_ $CavityID getDims]

```

```

set xminF [mins getValue 0 0 0]
set yminF [mins getValue 0 1 0]
set zminF [mins getValue 0 2 0]
set xmaxF [expr $xminF + [expr [expr double(1)/$NInterpUpsample]*[lindex
$InterpCavSize 0]]]
set ymaxF [expr $yminF + [expr [expr double(1)/$NInterpUpsample]*[lindex
$InterpCavSize 1]]]
set zmaxF [expr $zminF + [expr [expr double(1)/$NInterpUpsample]*[lindex
$InterpCavSize 2]]]

```

```

InterpCavVox_ $CavityID setBoundingBox $xminF $xmaxF $yminF $ymaxF $zminF
$zmaxF
InterpCavVox_ $CavityID\ _Wide setBoundingBox $xminF $xmaxF $yminF $ymaxF
$zminF $zmaxF

```

```

set IBox [InterpCavVox_ $CavityID getBoundingBox]

```

```

#Original bbox
set currCavBBox [Cav_Orig_ $specimen\_ $CavityID getBoundingBox]
set IntCavVoxSize [InterpCavVox_ $CavityID getVoxelSize]
set xmin [expr [expr [lindex $currCavBBox 0] -[lindex $IBox 0]]/[lindex
$IntCavVoxSize 0]]
set ymin [expr [expr [lindex $currCavBBox 2] -[lindex $IBox 2]]/[lindex
$IntCavVoxSize 1]]
set zmin [expr [expr [lindex $currCavBBox 4] -[lindex $IBox 4]]/[lindex
$IntCavVoxSize 2]]
set xmax [expr [expr [lindex $currCavBBox 1] - [lindex $IBox 0]] / [lindex
$IntCavVoxSize 0]]
set ymax [expr [expr [lindex $currCavBBox 3] - [lindex $IBox 2]] / [lindex
$IntCavVoxSize 1]]
set zmax [expr [expr [lindex $currCavBBox 5] - [lindex $IBox 4]] / [lindex
$IntCavVoxSize 2]]

```

```

InterpCavVox_ $CavityID crop $xmin $xmax $ymin $ymax $zmin $zmax

```

```

set cavDirectionCurr [expr round([mainDirection getValue 0 0 0])]
lappend cavDirection [expr round([mainDirection getValue 0 0 0])]
remove mainDirection2 vectors d locs mins2

```

```

set croppingCoords [list ]
lappend croppingCoords [expr [lindex $currCavBBox 0] -[lindex $OuterBBox 0]]
lappend croppingCoords [expr [lindex $currCavBBox 2] - [lindex $OuterBBox 2] ]
lappend croppingCoords [expr [lindex $currCavBBox 4] - [lindex $OuterBBox 4] ]

```

```

set mainDirection [mainDirection getValue 0 0 0]
set mainDirectionSigned [mainDirectionSigned getValue 0 0 0]
remove mainDirectionSigned mainDirectionSigned2 mainDirection mins

```

```

#Create surface of interpolated bone surface
create HxGMC CavSurfGen$CavityID
CavSurfGen$CavityID data connect InterpCavVox_ $CavityID\_Wide
CavSurfGen$CavityID smoothing setValue 0
CavSurfGen$CavityID options setValue border 1
CavSurfGen$CavityID create
remove CavSurfGen$CavityID

```

```

set currBBoxInterp [InterpCavVox_ $CavityID getBoundingBox]
set currCavBBox [VoxelCav_ $CavityID getBoundingBox]
set xmin1 [lindex $currCavBBox 0]
set ymin1 [lindex $currCavBBox 2]
set zmin1 [lindex $currCavBBox 4]
set xmin2 [lindex $currBBoxInterp 0]
set ymin2 [lindex $currBBoxInterp 2]
set zmin2 [lindex $currBBoxInterp 4]

```

```

set xmax1 [lindex $currCavBBox 1]
set ymax1 [lindex $currCavBBox 3]
set zmax1 [lindex $currCavBBox 5]
set xmax2 [lindex $currBBoxInterp 1]
set ymax2 [lindex $currBBoxInterp 3]
set zmax2 [lindex $currBBoxInterp 5]

```

CavDepth_VolandErrors.tcl

```

#Temporary method to get cav volume
while {[catch {[ create HxCalculus AdjustSpline ] getLabel} errmsg]} {
    create HxCalculus AdjustSpline }

```

```

AdjustSpline A connect InterpCavVox_ $CavityID
AdjustSpline A1 connect VoxelCav_ $CavityID
AdjustSpline A2 connect Cav$CavityID\_LowRes
AdjustSpline A3 connect Cav_Orig_ $specimen\_ $CavityID
#PairingMATLAB A1 connect dxBBox
AdjustSpline MatlabBuffer setValue "
%This code will resample a voxel image created from a surface
%gen of a cavity. Rather than directly resampling in 3d, it replicates
%slices to avoid errors from the staircase effect
%
%The code also corrects for spline surface errors due to peaks in the
%cavity passing through the spline surface
szCavOrig = size(A1);
szCavInt =size(A);
transO = \[round(($xmin2-$xmin1)/.7); round(($ymin2-$ymin1)/.7); round(($zmin2-
$zmin1)/5)\];
transI= \[round(($xmin2-$xmin1)/.5); round(($ymin2-$ymin1)/.5); round(($zmin2-
$zmin1)/.5)\];

tempOrig = uint8(zeros(szCavOrig (1)+abs(transO(1)),
szCavOrig(2)+abs(transO(2)),szCavOrig(3)));
tempInterp = uint8(zeros(szCavInt(1)+abs(transI(1)),
szCavInt(2)+abs(transI(2)),szCavInt(3)));
if transO(1) < 0 || transI(1)<0
    if transO(2) < 0 || transI(2)<0
        tempOrig(1+abs(transO(1)):end, 1:szCavOrig(2),:)=A1;
        tempInterp(1:szCavInt(1), 1+abs(transI(2)):end, :) =A;
    else
        tempOrig(1+abs(transO(1)):end, 1:szCavOrig(2),:)=A1;
        tempInterp(1:szCavInt(1), 1+abs(transI(2)):end,:) = A;
    end
end

else
    if transO(2) < 0 || transI(2)<0
        tempOrig(1:szCavOrig(1), 1+abs(transO(2)):end,:)= A1;
        tempInterp(1+abs(transI(1)):end, 1:szCavInt(2),:)=A;
    else
        tempOrig(1:szCavOrig(1), 1:szCavOrig(2),:)= A1;
        tempInterp(1+transI(1):end, 1+abs(transI(2)):end,:)= A;
    end
end

szCavInt = size(tempInterp);
imr=uint8(zeros(size(tempInterp)));
for i =1:szCavOrig(3)

```

```

    imr(:, :, i) = imresize(tempOrig(:, :, i), szCavInt(1:2));
end

```

```

NzUpsample = ceil(szCavInt(3)/szCavOrig(3));
imf = uint8(zeros(szCavInt(1), szCavInt(2), szCavOrig(3)*NzUpsample));
for i = 1:szCavOrig(3)
    for j = 1:NzUpsample
        imf(:, :, (i-1)*NzUpsample+j) = imr(:, :, i);
    end
end

```

```

imf = imf(:, :, 1:szCavInt(3));

```

```

if $mainDirection == 2
    mainDirection = 1;
elseif $mainDirection == 1
    mainDirection = 2;
else
    mainDirection = 3;
end

```

```

mask = logical(tempInterp);

```

```

for i = 1:szCavInt(3)

```

```

    locs = find(mask(:, :, i));
    [y, x] = ind2sub(szCavInt(1:2), locs);
    ymin = min(y);
    xmin = min(x);
    xmax = max(x);
    ymax = max(y);

```

```

    if mainDirection == 2
        mask(:, 1:xmin, i) = 1;
        mask(:, xmax:end, i) = 1;
        %extend interpolated BS to im boundary
        if $mainDirectionSigned == -1
            for k = 1:szCavInt(2)
                mask(1:find(mask(:, k, i)), k, i) = 1;
            end
        else
            for k = 1:szCavInt(2)

```

```

        mask(find(mask(:,k,i)):end,k,i) = 1;
    end
end

else %also allows for mainDirection in Z (may produce an erroroneous result)
    %however, cavities w a z normal are not (very rarely) labeled
    mask(1:ymin,:,i) = 1;
    mask(ymax:end,:,i) = 1;
    %extend interpolated BS to im boundary
    if $mainDirectionSigned== -1
        for k = 1:szCavInt(1)
            mask(k,1:find(mask(k, :, i)), i) = 1;
        end
    else
        for k = 1:szCavInt(1)
            mask(k,find(mask(k, :, i)):end, i) = 1;
        end
    end
end

mask(:, :, i) = imclose( mask(:, :, i), strel('disk', 2) );

end

CavVol = uint8((~mask.*logical(imf)))+tempInterp;
points2add=uint8(( uint8((mask.*logical(imclose(imf, strel('disk', 6))))))>0);

%Changed to 30 voxles on 6 10 2011, since this part is only for closing purposes
%d=$NInterpUpsample * ceil( *1.5);

%for i = 1:szCavInt(3)
    %t (d:d-1+szCavInt(1), d:d-1+szCavInt(2),i) = CavVol(:, :, i);
    % t (:, :, i) = imclose(t(:, :, i) , strel('disk', 30));

%end

%CavVol = uint8(t(d:d-1+szCavInt(1), d:d-1+szCavInt(2),:));

if sum(points2add(:)) ~=0
    adjust=1;
    MatlabExportList = { 'CavVol', 'points2add', 'adjust'};
else

```

```

adjust=0;
MatlabExportList = {'CavVol', 'adjust'};
end
"
AdjustSpline Apply snap 0 1
AdjustSpline fire
remove AdjustSpline
CavVol setLabel CavVol$CavityID

set BBoxi [VoxelCav_$CavityID getBoundingBox]
set CVsize [CavVol$CavityID getDims]

if {[expr $mainDirectionSigned<0]} {
  if {[expr $mainDirection == 1 ]} {

    set xmin [lindex $BBoxi 0]
    set xmax [expr $xmin + [expr [lindex $CVsize 0]/$NInterpUpsample]]
    set ymin [expr [lindex $BBoxi 2] + [expr $ymax2-$ymax1]]
    set ymax [expr $ymin + [expr [lindex $CVsize 1]/$NInterpUpsample]]
    set zmin [lindex $BBoxi 4]
    set zmax [expr $zmin + [expr [lindex $CVsize 2]/$NInterpUpsample]]

  } elseif {[expr $mainDirection == 2 ]} {

    set xmin [expr [lindex $BBoxi 0] + [expr $xmax2-$xmax1]]
    set xmax [expr $xmin + [expr [lindex $CVsize 0]/$NInterpUpsample]]
    set ymin [lindex $BBoxi 2]
    set ymax [expr $ymin + [expr [lindex $CVsize 1]/$NInterpUpsample]]
    set zmin [lindex $BBoxi 4]
    set zmax [expr $zmin + [expr [lindex $CVsize 2]/$NInterpUpsample]]
  } else {

    set xmin [expr [lindex $BBoxi 0] + [expr $xmin2-$xmin1]]
    set xmax [expr $xmin + [expr [lindex $CVsize 0]/$NInterpUpsample]]
    set ymin [expr [lindex $BBoxi 2] + [expr $ymin2-$ymin1]]
    set ymax [expr $ymin + [expr [lindex $CVsize 1]/$NInterpUpsample]]
    set zmin [lindex $BBoxi 4]
    set zmax [expr $zmin + [expr [lindex $CVsize 2]/$NInterpUpsample]]
  }

}

} else {

  set xmin [expr [lindex $BBoxi 0] + [expr $xmin2-$xmin1]]

```

```

set xmax [expr $xmin + [expr [lindex $CVsize 0]/$NInterpUpsample]]
set ymin [expr [lindex $BBoxi 2] + [expr $ymin2-$ymin1]]
set ymax [expr $ymin + [expr [lindex $CVsize 1]/$NInterpUpsample]]
set zmin [lindex $BBoxi 4]
set zmax [expr $zmin + [expr [lindex $CVsize 2]/$NInterpUpsample]]

}

```

CavDepthVols.tcl

```

#Temporary method to get cav volume
while {[catch {[ create HxCalculus AdjustSpline ] getLabel} errmsg]} {
    create HxCalculus AdjustSpline }

AdjustSpline A connect InterpCavVox_ $CavityID
AdjustSpline A1 connect VoxelCav_ $CavityID
AdjustSpline A2 connect Cav$CavityID\_LowRes
AdjustSpline A3 connect Cav_Orig_$specimen\_ $CavityID
AdjustSpline A4 connect d
#PairingMATLAB A1 connect dxBBox
AdjustSpline MatlabBuffer setValue "
%This code will resample a voxel image created from a surface
%gen of a cavity. Rather than directly resampling in 3d, it replicates
%slices to avoid errors from the staircase effect
%
%The code also corrects for spline surface errors due to peaks in the
%cavity passing through the spline surface

\[v,p\]=max(abs(mean(lineset.points')));
mainDirection = p;
m=(mean(lineset.points'));
mainDirectionSigned =sign(m(p));

szCavOrig = size(A1);
szCavInt =size(A);
transO = \[round(($xmin2-$xmin1)/.7); round(($ymin2-$ymin1)/.7); round(($zmin2-$zmin1)/5)\];
transI= \[round(($xmin2-$xmin1)/.5); round(($ymin2-$ymin1)/.5); round(($zmin2-$zmin1)/.5)\];

tempOrig = uint8(zeros(szCavOrig (1)+abs(transO(1)),
szCavOrig(2)+abs(transO(2)),szCavOrig(3)));

```



```

tempInterp = uint8(zeros(szCavInt(1)+abs(transI(1)),
szCavInt(2)+abs(transI(2)),szCavInt(3)));
if transO(1) < 0 || transI(1)<0
    if transO(2) < 0 || transI(2)<0
        tempOrig(1+abs(transO(1)):end, 1:szCavOrig(2),:)=A1;
        tempInterp(1:szCavInt(1), 1+abs(transI(2)):end, :) =A;
    else
        tempOrig(1+abs(transO(1)):end, 1:szCavOrig(2),:)=A1;
        tempInterp(1:szCavInt(1), 1+abs(transI(2)):end,:) = A;
    end
else
    if transO(2) < 0 || transI(2)<0
        tempOrig(1:szCavOrig(1), 1+abs(transO(2)):end,:)= A1;
        tempInterp(1+abs(transI(1)):end, 1:szCavInt(2),:)=A;
    else
        tempOrig(1:szCavOrig(1), 1:szCavOrig(2),:)= A1;
        tempInterp(1+transI(1):end, 1+abs(transI(2)):end,:)= A;
    end
end

szCavInt = size(tempInterp);
imr=uint8(zeros(size(tempInterp)));
for i =1:szCavOrig(3)
    imr(:,i)=imresize(tempOrig(:,i), szCavInt(1:2) );
end

NzUpsample = ceil(szCavInt(3)/szCavOrig(3));
imf=uint8(zeros(szCavInt(1), szCavInt(2), szCavOrig(3)*NzUpsample));
for i=1:szCavOrig(3)
    for j=1:NzUpsample
        imf(:,:(i-1)*NzUpsample+j) = imr(:,i);
    end
end

imf=imf(:,, 1:szCavInt(3));

if $mainDirection == 2
    mainDirection = 1;
elseif $mainDirection == 1
    mainDirection = 2;
end

```

```

mask = logical(tempInterp);

for i = 1:szCavInt(3)

    locs=find(mask(:,i));
    \[y,x\]=ind2sub(szCavInt(1:2), locs);
    ymin = min(y);
    xmin = min(x);
    xmax = max(x);
    ymax = max(y);

    if mainDirection == 2
        mask(1:xmin,i) = 1;
        mask(:,xmax:end, i) = 1;
        %extend interpolated BS to im boundary
        if $mainDirectionSigned== -1
            for k = 1:szCavInt(2)
                mask(1:find(mask(:,k, i)), k, i) = 1;
            end
        else
            for k = 1:szCavInt(2)
                mask(find(mask(:,k, i)):end,k, i) = 1;
            end
        end
    end

    else %also allows for mainDirection in Z (may produce an erroneous result)
        %however, cavities w a z normal are not (very rarely) labeled
        mask(1:ymin,:,i) = 1;
        mask(ymax:end,:,i) = 1;
        %extend interpolated BS to im boundary
        if $mainDirectionSigned== -1
            for k = 1:szCavInt(1)
                mask(k,1:find(mask(k, :, i)), i) = 1;
            end
        else
            for k = 1:szCavInt(1)
                mask(k,find(mask(k, :, i)):end, i) = 1;
            end
        end
    end
end

mask(:,i)=imclose( mask(:,i), strel('disk', 2) );

end

```

```

CavVol = uint8((~mask.*logical(imf))+tempInterp;

MatlabExportList = { 'CavVol'};
"
AdjustSpline Apply snap 0 1
AdjustSpline fire
remove AdjustSpline
CavVol setLabel CavVol$CavityID

set BBoxi [VoxelCav_$CavityID getBoundingBox]
set CVsize [CavVol$CavityID getDims]

if {[expr $mainDirectionSigned<0]} {
    if {[expr $mainDirection == 1 ] } {

        set xmin [lindex $BBoxi 0]
        set xmax [expr $xmin + [expr [lindex $CVsize 0]/$NInterpUpsample]]
        set ymin [expr [lindex $BBoxi 2] + [expr $ymax2-$ymax1]]
        set ymax [expr $ymin + [expr [lindex $CVsize 1]/$NInterpUpsample]]
        set zmin [lindex $BBoxi 4]
        set zmax [expr $zmin + [expr [lindex $CVsize 2]/$NInterpUpsample]]

    } elseif {[expr $mainDirection == 2 ] } {

        set xmin [expr [lindex $BBoxi 0] + [expr $xmax2-$xmax1]]
        set xmax [expr $xmin + [expr [lindex $CVsize 0]/$NInterpUpsample]]
        set ymin [lindex $BBoxi 2]
        set ymax [expr $ymin + [expr [lindex $CVsize 1]/$NInterpUpsample]]
        set zmin [lindex $BBoxi 4]
        set zmax [expr $zmin + [expr [lindex $CVsize 2]/$NInterpUpsample]]
    }

} else {

    set xmin [expr [lindex $BBoxi 0] + [expr $xmin2-$xmin1]]
    set xmax [expr $xmin + [expr [lindex $CVsize 0]/$NInterpUpsample]]
    set ymin [expr [lindex $BBoxi 2] + [expr $ymin2-$ymin1]]
    set ymax [expr $ymin + [expr [lindex $CVsize 1]/$NInterpUpsample]]
    set zmin [lindex $BBoxi 4]
    set zmax [expr $zmin + [expr [lindex $CVsize 2]/$NInterpUpsample]]

}

```

InterPointwrap.tcl

```

create HxPointWrap PW
PW data connect surf
PW fire
# Use whatever value is prescribed by amira initially
set ProbeRadInit [PW probeRadius getValue]
set ProbeRadMax [PW probeRadius getMaxValue]
set ProbeRad [expr $ProbeRadInit * 0.25]
PW probeRadius setValue $ProbeRad
# Use x-axis as search axis
PW searchAxis setValue 0
PW doIt snap
PW doIt send

```

```

set MinAcceptableArea [expr 0.75*[surf getArea]]

```

```

#If Initial PW failed

```

```

    #If the PW does not yet exist
    #SET SEARCH AXIS TO X
    if { [expr [string equal [PW getResult] {surf.surf}] != 1 ] } {
        set CurrentArea 0
        remove PW
        if { [expr $CurrentArea < $MinAcceptableArea] } {

            set axis 0
            remove PW
            set count0 0
            while { [expr [expr $CurrentArea < $MinAcceptableArea] &&
[expr $ProbeRad < $ProbeRadMax] && [expr $count0 < 100]] } {
                set count0 [expr $count0 + 1]
                echo $ProbeRad
            echo $CurrentArea
            create HxPointWrap PW
            PW data connect surf
            PW data send
            # Use 90% of max for probe radius
            set ProbeRad [expr $ProbeRad*1.05]
            PW probeRadius setValue $ProbeRad
            # Use x-axis as search axis
            PW searchAxis setValue $axis
            PW doIt snap
            PW doIt send

            if { [expr [string equal [PW getResult] {surf.surf}] != 1 ] } {

```

```

        set CurrentArea 0
    } else {
        set CurrentArea [surf.surf getArea]

    }
    remove PW
    remove surf.surf
}

#If the PW label exists, but is unreasonably small, continually reduce probe
sphere
} else {
    set ProbeRad $ProbeRadMax
    remove PW
    set CurrentArea [surf.surf getArea]
    if { [expr $CurrentArea < $MinAcceptableArea] } {
        set axis 0
        remove surf.surf

        set count1 0
        while { [expr [expr $CurrentArea < $MinAcceptableArea] &&
[expr $ProbeRad < $ProbeRadMax] && [expr $count1 < 100]] } {
            set count1 [expr $count1 + 1]

            echo $ProbeRad
            create HxPointWrap PW
            PW data connect surf
            PW data send
            # Use 90% of max for probe radius
            set ProbeRad [expr $ProbeRad*1.05]
            PW probeRadius setValue $ProbeRad
            # Use x-axis as search axis
            PW searchAxis setValue $axis
            PW doIt snap
            PW doIt send

            if {[catch {set CurrentArea [surf.surf getArea]}
errmsg]} {
                set CurrentArea 0
            } else {
                set CurrentArea [surf.surf getArea]
                remove surf.surf
                remove PW
            }
        }
    }
}

```

```

    }
}
remove surf.surf
set axis 0
#CREATE THE PW TO MAKE SURE IT EXISTS FOR SUBSEQUENT IF
STATEMENT
create HxPointWrap PW
PW data connect surf
PW data send
PW probeRadius setValue $ProbeRad
# Use x-axis as search axis
PW searchAxis setValue $axis
PW doIt snap
PW doIt send
remove PW

    if {[catch {set CurrentArea [surf.surf getArea]} errmsg]} {
        set CurrentArea 0
        create HxSurface
        Surface setLabel $axis
    } else {
        lappend Areas_ThinPW $CurrentArea
        surf.surf setLabel 0
    }

remove surf.surf

set ProbeRad $ProbeRadInit

#SET SEARCH AXIS TO Y
if {[catch {set CurrentArea [surf.surf getArea]} errmsg]} {
    set CurrentArea 0
    if { [expr $CurrentArea < $MinAcceptableArea] } {

        set axis 1
        remove surf.surf

        set count1 0
        while { [expr [expr $CurrentArea < $MinAcceptableArea] &&
[expr $ProbeRad < $ProbeRadMax] && [expr $count1 < 100]] } {
            set count1 [expr $count1 + 1]

            echo $ProbeRad
            create HxPointWrap PW
            PW data connect surf

```

```

PW data send
# Use 99% of max for probe radius
set ProbeRad [expr $ProbeRad*1.05]
PW probeRadius setValue $ProbeRad
# Use y-axis as search axis
PW searchAxis setValue $axis
PW doIt snap
PW doIt send
remove PW

if {[catch {set CurrentArea [surf.surf getArea]}
errmsg]} {
    set CurrentArea 0
} else {
    set CurrentArea [surf.surf getArea]
    remove surf.surf
}
}

#If the PW label exists, but is unreasonably small, continually reduce probe
sphere
} else {

set CurrentArea [surf.surf getArea]
remove surf.surf

if { [expr $CurrentArea < $MinAcceptableArea] } {
    set axis 1
    remove surf.surf

    set count1 0
    while { [expr [expr $CurrentArea < $MinAcceptableArea] &&
[expr $ProbeRad < $ProbeRadMax] && [expr $count1 < 100]] } {
        set count1 [expr $count1 + 1]
        echo $ProbeRad
        create HxPointWrap PW
        PW data connect surf
        PW data send
        # Use 90% of max for probe radius
        set ProbeRad [expr $ProbeRad*1.05]
        PW probeRadius setValue $ProbeRad
        # Use x-axis as search axis
        PW searchAxis setValue $axis
        PW doIt snap
        PW doIt send
    }
}

```

```

        remove PW

        if {[catch {set CurrentArea [surf.surf getArea]} errmsg]} {
            set CurrentArea 0
        } else {
            set Current Area [surf.surf getArea]
            remove surf.surf
        }

        remove surf.surf
    }
}

remove surf.surf

```

#CREATE THE PW TO MAKE SURE IT EXISTS FOR SUBSEQUENT IF STATEMENT

```

create HxPointWrap PW
PW data connect surf
PW data send
PW probeRadius setValue $ProbeRad
# Use x-axis as search axis
PW searchAxis setValue $axis
PW doIt snap
PW doIt send
remove PW

if {[catch {set CurrentArea [surf.surf getArea]} errmsg]} {
    set CurrentArea 0
    create HxSurface
    Surface setLabel $axis
} else {
    lappend Areas_ThinPW $CurrentArea
    surf.surf setLabel 1
}

remove surf.surf

set ProbeRad $ProbeRadInit
#SET SEARCH AXIS TO Z
if {[catch {set CurrentArea [surf.surf getArea]} errmsg]} {
    set CurrentArea 0
    if { [expr $CurrentArea < $MinAcceptableArea] } {

```



```

        set axis 2
        remove surf.surf

        set count1 0
        while { [expr [expr $CurrentArea < $MinAcceptableArea] &&
[expr $ProbeRad < $ProbeRadMax] && [expr $count1 < 100]] } {
            set count1 [expr $count1 + 1]
            echo $ProbeRad
            create HxPointWrap PW
            PW data connect surf
            PW data send
            # Use 99% of max for probe radius
            set ProbeRad [expr $ProbeRad*1.05]
            PW probeRadius setValue $ProbeRad
            # Use y-axis as search axis
            PW searchAxis setValue $axis
            PW doIt snap
            PW doIt send
            remove PW

            if {[catch {set CurrentArea [surf.surf getArea]}
errmsg]} {

                set CurrentArea 0
            } else {
                set CurrentArea [surf.surf getArea]
                remove surf.surf
            }
        }
    }

    #If the PW label exists, but is unreasonably small, continually reduce probe
sphere
    } else {

        set CurrentArea [surf.surf getArea]
        remove surf.surf

        if { [expr $CurrentArea < $MinAcceptableArea] } {
            set axis 2
            remove surf.surf

            set count1 0
            while { [expr [expr $CurrentArea < $MinAcceptableArea] &&
[expr $ProbeRad < $ProbeRadMax] && [expr $count1 < 100]] } {
                set count1 [expr $count1 + 1]
                echo $ProbeRad

```

```

        create HxPointWrap PW
        PW data connect surf
        PW data send
        # Use 90% of max for probe radius
        set ProbeRad [expr $ProbeRad*1.05]
        PW probeRadius setValue $ProbeRad
        # Use x-axis as search axis
        PW searchAxis setValue $axis
        PW doIt snap
        PW doIt send
        remove PW

        if {[catch {set CurrentArea [surf.surf getArea]} errmsg]} {
            set CurrentArea 0
        } else {
            set CurrentArea [surf.surf getArea]
            remove surf.surf
        }

        remove surf.surf
    }
}

```

#CREATE THE PW TO MAKE SURE IT EXISTS FOR SUBSEQUENT IF STATEMENT

```

create HxPointWrap PW
PW data connect surf
PW data send
PW probeRadius setValue $ProbeRad
# Use x-axis as search axis
PW searchAxis setValue $axis
PW doIt snap
PW doIt send
remove PW

if {[catch {set CurrentArea [surf.surf getArea]} errmsg]} {
    set CurrentArea 0
    create HxSurface
    Surface setLabel $axis
} else {
    lappend Areas_ThinPW $CurrentArea
    surf.surf setLabel 2
}

```

```
remove surf.surf
```

```
set A [list]
if {[catch {set CurrentArea [0 getArea]} errmsg] } {
lappend A 0
} else { lappend A [0 getArea] }
```

```
if {[catch {set CurrentArea [1 getArea]} errmsg] } {
lappend A 0
} else { lappend A [1 getArea] }
```

```
if {[catch {set CurrentArea [2 getArea]} errmsg] } {
lappend A 0
} else { lappend A [2 getArea] }
```

```
#USE MAXIMUM PW Area as the final PW'd surface
set CurrentPWArea [lindex [lsort -real $A] 2]
set CurrentPWSearchAxis [lsearch $A $CurrentPWArea]
```

```
$CurrentPWSearchAxis setLabel surf_PW
```

```
remove 0 1 2
remove surf
```

GetCavityAreas.tcl

```
#GET CAVITY Areas FOR each cavity
```

```
set desktopDir "Q:/"
set GlobalDir
"$desktopDir/uDx/$specimen\ROI/RESORPTION/Cavities/Get_Only_Area_From_The
e/"
set dataDir "$desktopDir\uDx/DATA/"
set cavDir $GlobalDir
```

```
#Find cavity files for current slice of interest
cd $GlobalDir
```

```
for {set SOI 1} {$SOI<=3} {incr SOI} {
```

```
if { [catch {set filenames [glob "*TraceCav_SOI$SOI*.am"]} ] } {
```

```

} else {

set NCavs [llength $filenames]

for {set i 1} {$i <= $NCavs} {incr i} {

    set currCavName [[load [lindex $filenames [expr $i-1]]] getLabel]

    set Acurr [$currCavName getArea]

    set TclLoadName [glob "SOI_$SOI*.tcl"]
    #Determine slices loaded for labeling current cavity
    source $GlobalDir[lindex $TclLoadName [expr $i-1]]

    #Remove all remaining variables before next loop iteration

    set outfile [open "$dataDir$specimen\_CavityData_Areas_Only.txt" a]
    puts $outfile "$SOI\t\t$CavityID\t\t\t$Acurr"
    close $outfile

    remove -all

}
}
}

```

SaveCav.tcl

```

#Amira script

#Run script to save files and output cavity surface area to text file w/ ID
set ImageOutDir "Q:/uDx/$specimen\ROI/RESORPTION/Cavities/"
file mkdir $ImageOutDir
set DataOutDir "Q:/uDx/DATA/"

create HxLineSet
LineSet addPoint [lindex $Scropcoords 0] [lindex $Scropcoords 2] [lindex $Scropcoords 4]
LineSet addPoint [lindex $Scropcoords 1] [lindex $Scropcoords 3] [lindex $Scropcoords 5]

```

```
LineSet addPoint $StartS0 $EndS0 0
```

```
$specimen\_i.surf setNoRename 1  
LineSet setNoRename 1
```

```
#Save image data (surfaces, etc)  
$specimen\_i.surf save "HxSurface binary"  
"$ImageOutDir/$specimen\_TraceCav\_SOI$SOI\_i.am"  
LineSet save "AmiraMesh ASCII"  
"$ImageOutDir/$specimen\_SOI$SOI\_i.GlobalCoords.am"
```

```
$specimen\_i.surf fire  
set box [$specimen\_i.surf getBoundingBox]
```

```
#Append cavity info text file with the bbox of the labeled cavity  
set outfile [open  
"Q:/uDx/$specimen/ROI/RESORPTION/Cavities/SOI_$SOI\_StartS$i.tcl" a]  
puts $outfile "  
set StartSLabeling $StartS0  
set EndSLabeling $EndS0  
set SOI $SOI  
set box \[list $box\  
set Grown $Grown  
set CavityID $i  
"  
close $outfile
```

growroi.tcl

```
#Amira Script  
#For cavity characterization - 6 / 9 / 11  
#Grow ROI .tcl will crop a slightly larger ROI if the current one is too small
```

```
remove -all
```

```
set desktopDir "Q:/"
```

```
set ROIDir "$desktopDir/uDx/$specimen/ROI/"  
set CoordPath "$desktopDir/uDx/$specimen/ROI/RESORPTION/CavityBoxes/"  
set resDir "$desktopDir/uDx/$specimen/ROI/RESORPTION/"
```

```

set EdgePath "$desktopDir\uDx\$specimen\ROI/Edge/"
set ThreshPath "$desktopDir\uDx\$specimen\ROI/Thresh/"
set GrayPath "$desktopDir\uDx\$specimen\ROI/UV/"

set Grown [expr $Grown +1]

#Specified variables
#Grayscale bone image path at 0.7 um
set ImPath "$ROIDir"
set OutDir "$ROIDir\RESORPTION/Cv_Ar_$SOI/Grown $Grown\"
file mkdir $OutDir

cd $CoordPath
set f [glob "$specimen\*ropping*oords*.tif"]
#Load cropping coordinates for slice of interest
set name [[load -tif +box 0 1 0 1 0 1 +mode 2 [lindex $f [expr $SOI-1]]] getLabel]
$name setLabel CroppingCoords

#How many microns to expand
set exp 150
set Nslices 800

#CALCULATED VARIABLES

#LOAD IMAGES
cd $GrayPath
set imageName [glob "*UV_ROI.tif"]

set imageName [lindex $imageName 0]
set initIm "$GrayPath/$imageName"
load -tif +box 0 1 0 1 0 1 +mode 2 $initIm
$imageName setLabel test
set imagesize [test getDims]

remove test

#Expand grayscale coordinates
set expXY [expr $exp/0.7]
set expZ [expr $exp/5]

```

```

#Determine slices to load
set StartS      [expr $StartS0 - $expZ*$Grown]
set EndS        [expr $EndS0 + $expZ*$Grown]

#Grayscale BBox coordinates
set xmin [expr int(ceil([expr [CroppingCoords getValue 0 [expr $i - 1 ] 0] -
$Grown*$expXY]))]
set ymin [expr int(ceil([expr [CroppingCoords getValue 1 [expr $i - 1 ] 0] -
$Grown*$expXY]))]
set xmax [expr int(ceil([expr [CroppingCoords getValue 2 [expr $i - 1 ] 0] +
$Grown*$expXY]))]
set ymax [expr int(ceil([expr [CroppingCoords getValue 3 [expr $i - 1 ] 0] +
$Grown*$expXY]))]

#Check for cropping images
if {[expr $xmin < 0]} {set xmin 0}
if {[expr $ymin < 0]} {set ymin 0}
if {[expr $StartS <= 0]} {
    set StartS 0
    echo "This cavity may extend above slice 0, respond to dialog box to continue"

}
if {[expr $xmax >= [lindex $imagesize 0]]} {set xmax [expr [lindex $imagesize 0]-1]}
if {[expr $ymax >= [lindex $imagesize 1]]} {set ymax [expr [lindex $imagesize 1]-1]}
if {[expr $EndS >= $Nslices]} {
    set EndS $Nslices
    echo "This cavity may extend below slice 800, respond to dialog box to continue"

}

#Save new coordinates to text file
set outfile [open "$OutDir/$specimen\_BS\_New\_BBox\_Grown.txt" a]
puts $outfile "
0.7 micron bounding box Grown $Grown
time(s)\t\t$xmin\t\t$xmax\t\t$ymin\t\t$ymax\t\t$StartS\t\t$EndS
"
close $outfile

set Nims [expr $EndS - $StartS +1]
cd $GrayPath

```

```

# ~~~~~LOAD
IMAGES~~~~~
set images [glob "*UV_ROI.tif"]
set images [lrange $images $StartS $EndS]
eval load -tif +box 0 [expr 0.7*([lindex $imagesize 0]-1)] 0 [expr 0.7*([lindex
$imagesize 1]-1)] 0 [expr 5*($Nims-1)] +mode 2 $images
[lindex $images 0] setLabel $specimen\_Gray$i
#Crop Image Stacks
$specimen\_Gray$i crop $xmin $xmax $ymin $ymax 0 [expr $Nims - 1]

cd $ThreshPath
set images [glob "*hresh_ROI.tif"]
set images [lrange $images $StartS $EndS]
eval load -tif +box 0 [expr 0.7*([lindex $imagesize 0]-1)] 0 [expr 0.7*([lindex
$imagesize 1]-1)] 0 [expr 5*($Nims-1)] +mode 2 $images
[lindex $images 0] setLabel $specimen\_Thresh$i
$specimen\_Thresh$i crop $xmin $xmax $ymin $ymax 0 [expr $Nims-1]

remove CroppingCoords

set crop_curr [list $xmin $xmax $ymin $ymax $StartS $EndS]

create HxOrthoSlice
OrthoSlice data connect $specimen\_Gray$i
OrthoSlice fire
OrthoSlice sliceNumber setValue 1000
OrthoSlice fire

#Create Surface of Thresh
create HxGMC SurfGen
SurfGen data connect $specimen\_Thresh$i
SurfGen smoothing setValue 2
SurfGen options setValue border 0
SurfGen create
remove SurfGen
$specimen\_Thresh$i\.surf setLabel $specimen\_ $i\.surf
$specimen\_ $i\.surf setNoRename 1

```



```

$specimen\_ $i\surf setNoRename 1
create HxDisplaySurface
SurfaceView setLabel showFullRes_$i
showFullRes_$i data connect $specimen\_ $i\surf
showFullRes_$i fire

#save
set netdir "$ROIDir\RESORPTION/Cv_Ar_$SOI/Grown $Grown\"
$specimen\_ $i\surf setNoRename 1
$specimen\_ $i\surf save "HxSurface binary"
"$netdir/$specimen\_SOI_$SOI\_Cv_$i\_Surf.am"
$specimen\_ Gray$i setNoRename 1
$specimen\_ Gray$i save "AmiraMesh ZIP"
"$netdir/$specimen\_SOI_$SOI\_Cv_$i\_Grayscale.am"
$specimen\_ Thresh$i setNoRename 1
$specimen\_ Thresh$i save "AmiraMesh ZIP"
"$netdir/$specimen\_SOI_$SOI\_Cv_$i\_Thresh.am"

set cropcoords [list $xmin $xmax $ymin $ymax $StartS $EndS]

#Save networks for each Cavity
set outfile [open
"$ROIDir\RESORPTION/SOI_$SOI\_CvAr_$i\_ROI_Grown_$Grown.hx" a]
puts $outfile "
# Amira Script
remove -all
viewer 0 setSize 1300 1000

create HxArithmetic
Arithmetic setLabel $specimen\_SOI_$SOI\_Cavity_$i

\[load \"$resDir/Cv_Ar_$SOI/Grown
$Grown/$specimen\_SOI_$SOI\_Cv_$i\_Surf.am\" \] setLabel $specimen\_ $i\surf
\[load \"$resDir/Cv_Ar_$SOI/Grown
$Grown/$specimen\_SOI_$SOI\_Cv_$i\_Surf.am\" \] setLabel
$specimen\_ $i\surfORIGINAL
\[load \"$resDir/Cv_Ar_$SOI/Grown
$Grown/$specimen\_SOI_$SOI\_Cv_$i\_Grayscale.am\" \] setLabel
$specimen\_ $i\_Grayscale
\[load \"$resDir/Cv_Ar_$SOI/Grown
$Grown/$specimen\_SOI_$SOI\_Cv_$i\_Thresh.am\" \] setLabel Thresh

create HxDisplaySurface
SurfaceView setLabel ShowFullRes
ShowFullRes data connect $specimen\_ $i\surf

```

ShowFullRes fire

```
create HxOrthoSlice
OrthoSlice data connect $specimen\_Si\_Grayscale
OrthoSlice sliceNumber setValue 10000
OrthoSlice transparency setValue 0
OrthoSlice linearRange setValues 0 \[expr \[OrthoSlice linearRange getValue 1\]/2\]
OrthoSlice fire
OrthoSlice select
```

```
set i $i
set specimen $specimen
set SOI $SOI
set Grown $Grown
set StartS0 $StartS
set EndS0 $EndS
set cropcoords [list $cropcoords ]
cd $desktopDir\uDx/scripts
"
close $outfile
```

```
#Save starting and ending slices to text file to establish coordinate system during depth
measure
set outfile [open "$resDir\Cv_Ar_$SOI\SOI_$SOI\_StartS_$i\.tcl" a]
puts $outfile "
set StartSLabeling $StartS
set EndSLabeling $EndS
set StartS0 $StartS
set EndS0 $EndS
set SOI $SOI
"
close $outfile
```

```
create HxUniformLabelField3
MS-Study setLabel $specimen\_SOI_$SOI\_Cavity\_Si\___GROWN_$Grown\_TIMTES

cd $desktopDir\uDx/scripts
```

Load_Ims_CvAr.tcl

#Creates a network with the grayscale images used to ID Cv s and measure CvAr

```

if ![info exists BATCH] {
    clear
    echo FORMAT:
    echo LoadCvAr <specimen> <StackNumber> <SOI>
}

proc LoadCvAr {specimen StackNumber SOI} {

    set StartS [expr SOI-150]
    set EndS [expr SOI+150]

    #Load stack pertaining to inputs

    set path "Q:/uDx/$specimen\ROI/RESORPTION/"
    cd $path

    set imageName [glob "*Gray*.tif"]
    set Nims [llength $imageName]
    set imageName [lindex $imageName 0]
    set initIm "$path/$imageName"
    load -tif +box 0 1 0 1 0 1 +mode 2 $initIm
    $imageName setLabel test
    set imageSize [test getDims]
    echo \tX-DIM: [lindex $imageSize 0] \t Y-DIM: [lindex $imageSize 1] \t Z-DIM: $Nims
    remove test

    # ~~~~~LOAD
    IMAGES~~~~~
    set images [glob "*Gray*.tif"]
    set images [lrange $images $StartS $EndS]
    eval load -tif +box 0 [expr 0.7*([lindex $imageSize 0]-1)] 0 [expr 0.7*([lindex
    $imageSize 1]-1)] 0 [expr 5*($Nims-1)] +mode 2 $images
    [lindex $images 0] setLabel $specimen\Gray

    set images [glob "*Thresh*.tif"]
    set images [lrange $images $StartS $EndS]
    eval load -tif +box 0 [expr 0.7*([lindex $imageSize 0]-1)] 0 [expr 0.7*([lindex
    $imageSize 1]-1)] 0 [expr 5*($Nims-1)] +mode 2 $images
    [lindex $images 0] setLabel $specimen\Thresh

    viewer 0 setCameraType 0
    create HxOrthoSlice
    OrthoSlice data connect $specimen\Gray
    OrthoSlice fire

```

```
OrthoSlice sliceNumber setValue [expr $Nims/2 - 1]
OrthoSlice fire
```

```
$specimen\Gray setLabel $specimen\Gray1
$specimen\Gray1 duplicate
$specimen\Gray setLabel $specimen\Gray1
$specimen\Gray2 setLabel $specimen\Gray_orig
}
```

Cavity Characterization Crop.tcl

```
#Code to crop grayscale cubes around cavities ID'd in 2D and
#to create surface of bone within those cubes
#Saves Networks of original lines, grayscale stacks, and surfaces
#Save text files with BBox locations
#SOI is the slice of interest
remove -all

set SOI 1
set i 1

set desktopDir "Q:/"

set scriptDir "$desktopDir\uDx/scripts/"
set funcDir "$desktopDir\uDx/scripts/functions/"
set spatialDir "$desktopDir\uDx/$specimen\ROI/SPATIAL/"
set dataDir "$desktopDir\uDx/DATA/"
set ROIDir "$desktopDir\uDx/$specimen\ROI/"
set specimenDir "$desktopDir\uDx/$specimen/"
set PTDPath "$spatialDir\PTDImages/"
set resDir "$ROIDir\RESORPTION/"

set EdgePath "$desktopDir\uDx/$specimen\ROI/Edge/"
set ThreshPath "$desktopDir\uDx/$specimen\ROI/Thresh/"
set GrayPath "$desktopDir\uDx/$specimen\ROI/UV/"

#Initiatlize parameters
set OrigRes [list 0.7 0.7 5]
set NSlices 800
#Number of slices to laod above and below current slice of interest
set Zdiff 40

set CoordPath "$resDir\CavityBoxes\\"
#Filenames for cropping coordinates
```

```

cd $CoordPath
set f [glob "$Specimen\*ropping*oords*.tif"]

if {      [expr [string equal $specimen C1] ] } { set slices [list 161 411 611]
} elseif {[expr [string equal $specimen C2] ] } { set slices [list 108 358 608]
} elseif {[expr [string equal $specimen C3] ] } { set slices [list 123 373 623]
} elseif {[expr [string equal $specimen C4] ] } { set slices [list 149 399 649]
} elseif {[expr [string equal $specimen C5] ] } { set slices [list 113 363 613]
} elseif {[expr [string equal $specimen C6] ] } { set slices [list 137 387 637]
} elseif {[expr [string equal $specimen C7] ] } { set slices [list 147 397 647]
} elseif {[expr [string equal $specimen C8] ] } { set slices [list 123 373 623]
} elseif {[expr [string equal $specimen C9] ] } { set slices [list 148 398 648]}

#For each slice, save VOI around each cavity
for {set SOI 1} {$SOI<=3} {incr SOI} {
    set netdir "$resDir/Cv_Ar_$SOI/"
    file mkdir $netdir

    cd $CoordPath
    #Load cropping coordinates for slice of interest
    set name [[load -tif +box 0 1 0 1 0 1 +mode 2 [lindex $f [expr $SOI-1]]]
getLabel]
    $name setLabel CroppingCoords
    set NCavs [lindex [CroppingCoords getDims] 1]

    #Load in first grayscale to determine size
    cd $GrayPath
    set imageNameGray [glob "*UV_ROI.tif"]
    set imageNameGray [lindex $imageNameGray 0]
    set initIm "$GrayPath/$imageNameGray"
    load -tif +box 0 1 0 1 0 1 +mode 2 $initIm
    $imageNameGray setLabel test
    set sz [test getDims]
    remove test

    #Determine slices to load
    set StartS      [expr [lindex $slices [expr $SOI - 1 ]] - $Zdiff ]
    set EndS        [expr [lindex $slices [expr $SOI - 1 ]] + $Zdiff ]

    echo "Load all grayscale images in StartS - EndS for SOI $SOI"
    set filenamesGray [glob "*UV_ROI.tif"]
    set filenamesGray "$GrayPath/$filenamesGray"
    set Nims [expr $EndS - $StartS +1]
    echo "loading $Nims images for cavity slice $SOI"
    set filenamesGray [lrange $filenamesGray $StartS $EndS]

```

```

    set name [[eval load -tif +box 0 [expr 0.7*([lindex $sz 0]-1)] 0 [expr 0.7*([lindex
$sz 1]-1)] 0 [expr 5*($Nims-1)] +mode 2 $filenamesGray] getLabel]
    $name setLabel Gray
    set imagesize [Gray getDims]

```

```

    cd $ThreshPath
    echo "Load all threshold images in StartS - EndS for SOI $SOI"
    set filenamesThresh [glob "*Thresh*0p7.tif"]
    set filenamesThresh "$ThreshPath/$filenamesThresh"
    set filenamesThresh [lrange $filenamesThresh $StartS $EndS]
    set name [[eval load -tif +box 0 [expr 0.7*([lindex $sz 0]-1)] 0 [expr 0.7*([lindex
$sz 1]-1)] 0 [expr 5*($Nims-1)] +mode 2 $filenamesThresh ] getLabel]
    $name setLabel Thresh

```

#For each cavity in slice, save VOI and Amira network

```

for {set i 1} {$i<=$NCavs} {incr i} {
    echo "Cropping images for cavity labeling $i of $NCavs"
    #Grayscale BBox coordinates (stored in 16 bit tiff format)
    set xmin      [expr [CroppingCoords getValue 0 [expr $i - 1 ] 0] - 50]
    set ymin      [expr [CroppingCoords getValue 1 [expr $i - 1 ] 0] -50]
    set xmax      [expr [CroppingCoords getValue 2 [expr $i - 1 ] 0]-50]
    set ymax      [expr [CroppingCoords getValue 3 [expr $i - 1 ] 0]-50]

    #Check that slices are within the stack (ie 0 - 800)
    if {[expr $xmin < 0]} { set xmin 0}
    if {[expr $ymin < 0]} {set ymin 0}
    if {[expr $xmax > [lindex $sz 0] ]} { set xmax [lindex $sz 0]}
    if {[expr $ymax > [lindex $sz 1] ]} {set ymax [lindex $sz 1]}

    #Duplicate and crop gray and thresh images to cavity bbox
    set nGray [[Gray duplicate] getLabel]
    $nGray setLabel $specimen\_Gray$i
    $specimen\_Gray$i crop $xmin $xmax $ymin $ymax 0 [expr [lindex $imagesize
2]- 1]

```

```

    set nThresh [[Thresh duplicate] getLabel]
    $nThresh setLabel $specimen\_Thresh$i
    $specimen\_Thresh$i crop $xmin $xmax $ymin $ymax 0 [expr [lindex
$imagesize 2]- 1]

```

```

    #Create bone surface to label cavity on
    create HxGMC SurfGen
    SurfGen data connect $specimen\_Thresh$i
    SurfGen smoothing setValue 0
    SurfGen options setValue border 0
    SurfGen create

```

```

remove SurfGen
$specimen\_Thresh$i\.surf setLabel $specimen\_i\.surf
$specimen\_i\.surf setNoRename 1

#save
echo "saving grayscale and surface files"
$specimen\_i\.surf save "HxSurface binary"
"$netdir/$specimen\_SOI_$SOI\_Cv\_i\_Surf.am"
$specimen\_Gray$i setNoRename 1
$specimen\_Gray$i save "AmiraMesh ZIP"
"$netdir/$specimen\_SOI_$SOI\_Cv\_i\_Grayscale.am"
$specimen\_Thresh$i setNoRename 1
$specimen\_Thresh$i save "AmiraMesh ZIP"
"$netdir/$specimen\_SOI_$SOI\_Cv\_i\_Thresh.am"

set cropcoords [list $xmin $xmax $ymin $ymax $StartS $EndS]
#Save networks for each Cavity
set outfile [open "$resDir/SOI_$SOI\_CvAr\_i\.hx" a]
puts $outfile "
# Amira Script
remove -all
viewer 0 setSize 1300 1000

create HxUniformLabelField3
MS-Study setLabel $specimen\_SOI_$SOI\_Cavity\_i

\load \"$resDir/Cv\_Ar\_SOI/$specimen\_SOI_$SOI\_Cv\_i\_Surf.am\" \]
setLabel $specimen\_i\.surf
\load \"$resDir/Cv\_Ar\_SOI/$specimen\_SOI_$SOI\_Cv\_i\_Surf.am\" \]
setLabel $specimen\_i\.surfORIGINAL
\load \"$resDir/Cv\_Ar\_SOI/$specimen\_SOI_$SOI\_Cv\_i\_Grayscale.am\"
\] setLabel $specimen\_i\_Grayscale
\load \"$resDir/Cv\_Ar\_SOI/$specimen\_SOI_$SOI\_Cv\_i\_Thresh.am\" \]
setLabel Thresh

create HxDisplaySurface
SurfaceView setLabel ShowFullRes
ShowFullRes data connect $specimen\_i\.surf
ShowFullRes fire
ShowFullRes drawStyle setValue 1
ShowFullRes fire

create HxOrthoSlice
OrthoSlice data connect $specimen\_i\_Grayscale
OrthoSlice sliceNumber setValue 10000

```

```

OrthoSlice transparency setValue 0
OrthoSlice linearRange setValues 0 \[expr \[OrthoSlice linearRange getValue
1\]/2\]
OrthoSlice fire
OrthoSlice select

set i $i
set specimen $specimen
set SOI $SOI
set Grown 0
set StartS0 $StartS
set EndS0 $EndS
set cropcoords [list $cropcoords]
cd $desktopDir\uDx/scripts

"
close $outfile

#Save starting and ending slices to text file to establish coordinate system during
depth measure
set outfile [open "$resDir\Cv_Ar_$SOI\SOI_$SOI\_StartS_$i\.tcl" a]
puts $outfile "
set StartSLabeling $StartS
set EndSLabeling $EndS
set StartS0 $StartS
set EndS0 $EndS
set SOI $SOI
"
close $outfile

remove $specimen\_Gray$i $specimen\_Thresh$i $specimen\_i\.surf
}
remove Gray Thresh CroppingCoords
}

```


REFERENCES

1. Gallagher, J.C. and A. Sai, *Molecular biology of bone remodeling: Implications for new therapeutic targets for osteoporosis*. Maturitas, 2010. **65**(4): p. 301-307.
2. Cooper, C., *Epidemiology of osteoporosis*, in *Primer on the metabolic bone diseases and disorders of mineral metabolism*, M.J. Favus, Editor 2003, American society for bone and mineral research: Washington, DC, USA. p. 307-313.
3. Cooper, C. and L.J. Melton, 3rd, *Magnitude and impact of osteoporosis and fractures*, in *Osteoporosis*, R. Marcus, D. Feldman, and J.L. Kelsey, Editors. 1996, Academic Press: San Diego. p. 419-434.
4. Melton, L.J., et al., *Long-term fracture prediction by bone mineral assessed at different skeletal sites*. Journal of Bone and Mineral Research, 1993. **8**(10): p. 1227-1233.
5. Marshall, D., O. Johnell, and H. Wedel, *Meta-Analysis of how well measures of bone mineral density predict occurrence of osteoporotic fractures*. BMJ, 1996. **312**: p. 1254-59.
6. Schuit, S.C., et al., *Fracture incidence and association with bone mineral density in elderly men and women: the Rotterdam Study*. Bone, 2004. **34**(1): p. 195-202.
7. Delmas, P.D. and E. Seeman, *Changes in bone mineral density explain little of the reduction in vertebral or nonvertebral fracture risk with anti-resorptive therapy*. Bone, 2004. **34**(4): p. 599-604.
8. Riggs, B.L. and L.J. Melton, 3rd, *Bone turnover matters: the raloxifene treatment paradox of dramatic decreases in vertebral fractures without commensurate increases in bone density*. J Bone Miner Res, 2002. **17**(1): p. 11-4.
9. Cummings, S.R., et al., *Improvement in spine bone density and reduction in risk of vertebral fractures during treatment with antiresorptive drugs*. American Journal of Medicine, 2002. **112**(4): p. 281-289.

10. Parfitt, A.M., *Osteonal and hemi-osteonal remodeling: the spatial and temporal framework for signal traffic in adult human bone*. J Cell Biochem, 1994. **55**(3): p. 273-86.
11. Garnero, P., et al., *Biochemical markers of bone turnover, endogenous hormones and the risk of fractures in postmenopausal women: the OFELY study*. Journal of Bone and Mineral Research, 2000. **15**(8): p. 1526-1536.
12. Fazzalari, N.L., *Bone remodeling: A review of the bone microenvironment perspective for fragility fracture (osteoporosis) of the hip*. Semin Cell Dev Biol, 2008. **19**(5): p. 467-72.
13. van der Linden, J.C., et al., *Mechanical consequences of bone loss in cancellous bone*. J Bone Miner Res, 2001. **16**(3): p. 457-65.
14. Hernandez, C.J., A. Gupta, and T.M. Keaveny, *A biomechanical analysis of the effects of resorption cavities on cancellous bone strength*. J Bone Miner Res, 2006. **21**(8): p. 1248-55.
15. McNamara, L.M.v.d.L., J. C.; Weinans, H.; Prendergast, P. J. *Stress-concentrating effect of resorption lacunae in trabecular bone*. Journal of Biomechanics, 2006. **39**, 734-41.
16. Lian, J., *Bone formation: osteoblast lineage cells, growth factors, matrix proteins and the mineralization process*. Primer on the metabolic bone diseases and disorders of mineral metabolism, 1999.
17. Gorski, J., *Is all bone the same? Distinctive distributions and properties of non-collagenous matrix proteins in lamellar vs. woven bone imply the existence of different underlying osteogenic mechanisms*. Critical Reviews in Oral Biology & Medicine, 1998. **9**(2): p. 201.
18. Parfitt, A.M., et al., *Relationships between Surface, Volume, and Thickness of Iliac Trabecular Bone in Aging and in Osteoporosis - Implications for the Microanatomic and Cellular Mechanisms of Bone Loss*. Journal of Clinical Investigation, 1983. **72**(4): p. 1396-1409.

19. Williams, J.L. and W.J.H. Johnson, *Elastic constants of composites formed from PMMA bone cement and anisotropic bovine tibial cancellous bone*. Journal of biomechanics, 1989. **22**(6-7): p. 673-682.
20. Wolff, J., *Das Gesetz der Transformation der Knochen*. 1892, Berlin: Hirschwald.
21. Zysset, P.K., R.W. Goulet, and S.J. Hollister, *A global relationship between trabecular bone morphology and homogenized elastic properties*. Journal of Biomechanical Engineering, 1998. **120**(5): p. 640-6.
22. Martin, R.B. and D.L. Boardman, *The effects of collagen fiber orientation, porosity, density, and mineralization on bovine cortical bone bending properties*. Journal of Biomechanics, 1993. **26**(9): p. 1047-54.
23. Carter, D.R. and D.M. Spengler, *Mechanical properties and composition of cortical bone*. Clin. Orthop., 1978. **135**(135): p. 192-217.
24. Brown, T.D. and A.B. Ferguson, *Mechanical property distributions in the cancellous bone of the human proximal femur*. Acta Orthop Scand, 1980. **51**(3): p. 429-37.
25. Harrigan, T.P., et al., *Limitations of the continuum assumption in cancellous bone*. Journal of Biomechanics, 1988. **21**(4): p. 269-75.
26. Odgaard, A., et al., *Fabric and elastic principal directions of cancellous bone are closely related*. Journal of Biomechanics, 1997. **30**(5): p. 487-495.
27. Yang, G., et al., *The anisotropic Hooke's law for cancellous bone and wood*. Journal of Elasticity, 1999. **53**(2): p. 125-146.
28. Mosekilde, L., L. Mosekilde, and C.C. Danielsen, *Biomechanical competence of vertebral trabecular bone in relation to ash density and age in normal individuals*. Bone, 1987. **8**(2): p. 79-85.
29. Ciarelli, M.J., et al., *Evaluation of Orthogonal Mechanical-Properties and*

Density of Human Trabecular Bone from the Major Metaphyseal Regions with Materials Testing and Computed-Tomography. Journal of Orthopaedic Research, 1991. **9**(5): p. 674-682.

30. Williams, J.L. and J.L. Lewis, *Properties and an anisotropic model of cancellous bone from the proximal tibial epiphysis.* Journal of Biomedical Engineering, 1982. **104**: p. 50-56.
31. Cowin, S., ed. *Bone Mechanics Handbook.* 2 ed. 2001, CRC Press: Boca Raton. 968.
32. Kopperdahl, D.L. and T.M. Keaveny, *Yield strain behavior of trabecular bone.* Journal of Biomechanics, 1998. **31**(7): p. 601-8.
33. Linde, F., I. Hvid, and B. Pongsoipetch, *Energy absorptive properties of human trabecular bone specimens during axial compression.* Journal of Orthopaedic Research, 1989. **7**(3): p. 432-9.
34. Morgan, E.F., et al., *Nonlinear behavior of trabecular bone at small strains.* J Biomech Eng, 2001. **123**(1): p. 1-9.
35. Carter, D.R. and W.C. Hayes, *The compressive behavior of bone as a two-phase porous structure.* Journal of Bone and Joint Surgery, 1977. **59-A**: p. 954 - 962.
36. Rice, J.C., S.C. Cowin, and J.A. Bowman, *On the dependence of the elasticity and strength of cancellous bone on apparent density.* Journal of Biomechanics, 1988. **21**(2): p. 155-168.
37. Keaveny, T.M., et al., *Non-Linear elastic behavior of trabecular bone.* World Congress of Biomechanics, 1998. **3**: p. 428.
38. Lotz, J.C., T.N. Gerhart, and W.C. Hayes, *Mechanical properties of trabecular bone from the proximal femur: a quantitative CT study.* J Comput Assist Tomogr, 1990. **14**(1): p. 107-14.
39. Rohlmann, A., et al., *Material properties of femoral cancellous bone in axial*

loading. Part I: Time independent properties. Arch Orthop Trauma Surg, 1980. **97**(2): p. 95-102.

40. Mosekilde, L. and L. Mosekilde, *Normal vertebral body size and compressive strength: relations to age and to vertebral and iliac trabecular bone compressive strength.* Bone, 1986. **7**: p. 207-212.
41. Ding, M., et al., *Age variations in the properties of human tibial trabecular bone.* Journal of Bone and Joint Surgery. British Volume, 1997. **79**(6): p. 995-1002.
42. Keaveny, T.M., et al., *Application of the Tsai-Wu quadratic multiaxial failure criterion to bovine trabecular bone.* Journal of Biomechanical Engineering, 1999. **121**: p. 99-107.
43. Keaveny, T.M., et al., *Differences between the tensile and compressive strengths of bovine tibial trabecular bone depend on modulus.* Journal of Biomechanics, 1994. **27**: p. 1137-1146.
44. Keaveny, T.M., E.F. Wachtel, and D.L. Kopperdahl, *Mechanical behavior of human trabecular bone after overloading.* Journal of Orthopaedic Research, 1999. **17**: p. 346-53.
45. Morgan, E.F., O.C. Yeh, and T.M. Keaveny, *Damage in trabecular bone at small strains.* Eur J Morphol, 2005. **42**(1-2): p. 13-21.
46. Vashishth, D., et al., *In vivo diffuse damage in human vertebral trabecular bone.* Bone, 2000. **26**: p. 147-152.
47. Fazzalari, N.L., B. Vernon-Roberts, and J. Darracott, *Osteoarthritis of the hip: possible protective and causative roles of trabecular microfractures in the head of the femur.* Clinical orthopaedics and related research, 1987. **216**: p. 224.
48. Fyhrie, D.P. and M.B. Schaffler, *Failure mechanisms in human vertebral cancellous bone.* Bone, 1994. **15**(1): p. 105-9.
49. Burr, D.B., et al., *Bone microdamage and skeletal fragility in osteoporotic and*

- stress fractures*. Journal of Bone and Mineral Research, 1997. **12**(1): p. 6-15.
50. Bentalila, V., et al., *Intracortical remodeling in adult rat long bones after fatigue loading*. Bone, 1998. **23**(3): p. 275-81.
 51. Burr, D.B., et al., *Bone remodeling in response to in vivo fatigue microdamage*. Journal of Biomechanics, 1985. **18**(3): p. 189-200.
 52. Ross, F., et al., *Interactions between the bone matrix proteins osteopontin and bone sialoprotein and the osteoclast integrin α v β 3 potentiate bone resorption*. Journal of Biological Chemistry, 1993. **268**(13): p. 9901.
 53. Drake, F.H., et al., *Cathepsin K, but not cathepsins B, L, or S, is abundantly expressed in human osteoclasts*. Journal of Biological Chemistry, 1996. **271**(21): p. 12511.
 54. Bossard, M.J., et al., *Proteolytic activity of human osteoclast cathepsin K*. Journal of Biological Chemistry, 1996. **271**(21): p. 12517.
 55. Suda, T., N. Takahashi, and T. Martin, *Modulation of osteoclast differentiation*. Endocrine Reviews, 1992. **13**(1): p. 66.
 56. Saftig, P., et al., *Impaired osteoclastic bone resorption leads to osteopetrosis in cathepsin-K-deficient mice*. Proceedings of the National Academy of Sciences, 1998. **95**(23): p. 13453.
 57. Yoshida, H., et al., *The murine mutation osteopetrosis is in the coding region of the macrophage colony stimulating factor gene*. 1990.
 58. Jimi, E., et al., *Osteoclast differentiation factor acts as a multifunctional regulator in murine osteoclast differentiation and function*. The Journal of Immunology, 1999. **163**(1): p. 434.
 59. Burgess, T.L., et al., *The ligand for osteoprotegerin (OPGL) directly activates mature osteoclasts*. The Journal of cell biology, 1999. **145**(3): p. 527.

60. Fuller, K., et al., *TRANCE is necessary and sufficient for osteoblast-mediated activation of bone resorption in osteoclasts*. The Journal of experimental medicine, 1998. **188**(5): p. 997.
61. Lum, L., et al., *Evidence for a role of a tumor necrosis factor- (TNF-)-converting enzyme-like protease in shedding of TRANCE, a TNF family member involved in osteoclastogenesis and dendritic cell survival*. Journal of Biological Chemistry, 1999. **274**(19): p. 13613.
62. Miyauchi, A., et al., *Recognition of osteopontin and related peptides by an alpha v beta 3 integrin stimulates immediate cell signals in osteoclasts*. Journal of Biological Chemistry, 1991. **266**(30): p. 20369.
63. Chambers, T., J. Darby, and K. Fuller, *Mammalian collagenase predisposes bone surfaces to osteoclastic resorption*. Cell and tissue research, 1985. **241**(3): p. 671-675.
64. Hughes, D.E., et al., *Estrogen promotes apoptosis of murine osteoclasts mediated by TGF-[beta]*. Nat Med, 1996. **2**(10): p. 1132-1136.
65. Marotti, G., A. Zallone, and M. Ledda, *Number, size and arrangement of osteoblasts in osteons at different stages of formation*. Calcif Tiss Int, 1975. **21**(1): p. 96-101.
66. Eriksen, E.F., et al., *Cancellous bone remodeling in type I (postmenopausal) osteoporosis: quantitative assessment of rates of formation, resorption, and bone loss at tissue and cellular levels*. Journal of Bone and Mineral Research, 1990. **5**(4): p. 311-9.
67. Bradbeer, J.N., P.C. Lindsay, and J. Reeve, *Fluctuation of mineral apposition rate at individual bone-remodeling sites in human iliac cancellous bone: Independent correlations with osteoid width and osteoblastic alkaline phosphatase activity*. Journal of Bone and Mineral Research, 1994. **9**(11): p. 1679-1686.
68. Hauschka, P., et al., *Growth factors in bone matrix. Isolation of multiple types by affinity chromatography on heparin-Sepharose*. Journal of Biological Chemistry, 1986. **261**(27): p. 12665.

69. Glimcher, M.J., *Mechanism of calcification: Role of collagen fibrils and collagen phosphoprotein complexes in vitro and in vivo*. The Anatomical Record, 1989. **224**(2): p. 139-153.
70. Hanson, D.A. and D.R. Eyre, *Molecular site specificity of pyridinoline and pyrrole cross-links in type I collagen of human bone*. J Biol Chem, 1996. **271**(43): p. 26508-16.
71. Viguet-Carrin, S., P. Garnero, and P. Delmas, *The role of collagen in bone strength*. Osteoporosis International, 2006. **17**(3): p. 319-336.
72. Noble, B., et al., *Identification of apoptotic changes in osteocytes in normal and pathological human bone*. Bone, 1997. **20**(3): p. 273-282.
73. Burger, E.H. and J. Klein-Nulend, *Mechanotransduction in bone-role of the lacunocanalicular network*. Faseb Journal, 1999. **13**: p. 101-112.
74. Klein-Nulend, J., et al., *Sensitivity of osteocytes to biomechanical stress in vitro*. Faseb Journal, 1995. **9**(5): p. 441-5.
75. Weinstein, R.S., et al., *Inhibition of osteoblastogenesis and promotion of apoptosis of osteoblasts and osteocytes by glucocorticoids. Potential mechanisms of their deleterious effects on bone*. Journal of Clinical Investigation, 1998. **102**(2): p. 274.
76. Dalle Carbonare, L., et al., *Histomorphometric analysis of glucocorticoid-induced osteoporosis*. Micron, 2005. **36**(7-8): p. 645-652.
77. Verborgt, O., G.J. Gibson, and M.B. Schaffler, *Loss of osteocyte integrity in association with microdamage and bone remodeling after fatigue in vivo*. J Bone Miner Res, 2000. **15**(1): p. 60-7.
78. Vashishth, D., et al., *In vivo diffuse damage in human vertebral trabecular bone*. Bone, 2000. **26**(2): p. 147-152.

79. Fazzalari, N.L., *Trabecular microfracture*. Calcified Tissue International, 1993. **53**(S1): p. S143-S147.
80. Burr, D.B., *Remodeling and the repair of fatigue damage*. Calcified Tissue International, 1993: p. S75-80; discussion S80-1.
81. Jaworski, Z., M. Liskova-Kiar, and H. Uhthoff, *Effect of long-term immobilisation on the pattern of bone loss in older dogs*. Journal of Bone and Joint Surgery-British Volume, 1980. **62**(1): p. 104.
82. Parfitt, A.M., *The coupling of bone formation to bone resorption: a critical analysis of the concept and of its relevance to the pathogenesis of osteoporosis*. Metab Bone Dis Relat Res, 1982. **4**(1): p. 1-6.
83. Eriksen, E.F., et al., *Reconstruction of the formative site in iliac trabecular bone in 20 normal individuals. A kinetic model for matrix and mineral apposition*. Metab Bone Dis Relat Res, 1984. **5**: p. 243-252.
84. Frost, H.M. and C. Straatsma, *Bone remodelling dynamics*. Plastic and Reconstructive Surgery, 1964. **33**(2): p. 196.
85. Frost, H.M., *Bone dynamics in metabolic bone disease*. J Bone Joint Surg, 1966. **48**(6): p. 1192.
86. Parfitt, A.M., *Misconceptions (2): turnover is always higher in cancellous than in cortical bone*. Bone, 2002. **30**(6): p. 807-9.
87. Parfitt, A.M., *What is the normal rate of bone remodeling?* Bone, 2004. **35**(1): p. 1-3.
88. Garnero, P., et al., *Biochemical markers of bone turnover, endogenous hormones and the risk of fractures in postmenopausal women: The OFELY study*. Journal of Bone and Mineral Research, 2000. **15**(8): p. 1526-1536.
89. Kong, Y.Y., et al., *OPGL is a key regulator of osteoclastogenesis, lymphocyte*

development and lymph-node organogenesis. Nature, 1999. **397**(6717): p. 315-323.

90. Felix, R., M. Cecchini, and H. Fleisch, *Macrophage colony stimulating factor restores in vivo bone resorption in the op/op osteopetrotic mouse*. Endocrinology, 1990. **127**(5): p. 2592.
91. Kahn, A. and D. Simmons, *Investigation of cell lineage in bone using a chimaera of chick and quail embryonic tissue*. 1975.
92. Young, R.W., *Cell proliferation and specialization during endochondral osteogenesis in young rats*. The Journal of cell biology, 1962. **14**(3): p. 357.
93. Roodman, G., et al., *1, 25-Dihydroxyvitamin D3 causes formation of multinucleated cells with several osteoclast characteristics in cultures of primate marrow*. Proceedings of the National Academy of Sciences of the United States of America, 1985. **82**(23): p. 8213.
94. MacDonald, B., et al., *Effects of human recombinant CSF GM and highly purified CSF 1 on the formation of multinucleated cells with osteoclast characteristics in long term bone marrow cultures*. Journal of Bone and Mineral Research, 1986. **1**(2): p. 227-233.
95. Tamura, T., et al., *Soluble interleukin-6 receptor triggers osteoclast formation by interleukin 6*. Proceedings of the National Academy of Sciences, 1993. **90**(24): p. 11924.
96. Udagawa, N., et al., *Interleukin (IL)-6 induction of osteoclast differentiation depends on IL-6 receptors expressed on osteoblastic cells but not on osteoclast progenitors*. The Journal of experimental medicine, 1995. **182**(5): p. 1461.
97. Liu, B.Y., et al., *Conditionally immortalized murine bone marrow stromal cells mediate parathyroid hormone-dependent osteoclastogenesis in vitro*. Endocrinology, 1998. **139**(4): p. 1952.
98. Takeda, S., et al., *Stimulation of osteoclast formation by 1, 25-dihydroxyvitamin D requires its binding to vitamin D receptor (VDR) in osteoblastic cells: studies*

using VDR knockout mice. *Endocrinology*, 1999. **140**(2): p. 1005.

99. O'Brien, C.A., et al., *STAT3 activation in stromal/osteoblastic cells is required for induction of the receptor activator of NF- κ B ligand and stimulation of osteoclastogenesis by gp130-utilizing cytokines or interleukin-1 but not 1, 25-dihydroxyvitamin D3 or parathyroid hormone*. *Journal of Biological Chemistry*, 1999. **274**(27): p. 19301.
100. Simonet, W., et al., *Osteoprotegerin: a novel secreted protein involved in the regulation of bone density*. *Cell*, 1997. **89**(2): p. 309-319.
101. Yasuda, H., et al., *Osteoclast differentiation factor is a ligand for osteoprotegerin/osteoclastogenesis-inhibitory factor and is identical to TRANCE/RANKL*. *Proceedings of the National Academy of Sciences*, 1998. **95**(7): p. 3597.
102. Quinn, J.M.W., et al., *Transforming growth factor affects osteoclast differentiation via direct and indirect actions*. *Journal of Bone and Mineral Research*, 2001. **16**(10): p. 1787-1794.
103. Fox, S.W., et al., *TGF- β 1 and IFN- γ Direct Macrophage Activation by TNF- α to Osteoclastic or Cytocidal Phenotype*. *The Journal of Immunology*, 2000. **165**(9): p. 4957.
104. Robey, P.G., et al., *Osteoblasts synthesize and respond to transforming growth factor-type beta (TGF-beta) in vitro*. *The Journal of cell biology*, 1987. **105**(1): p. 457.
105. Pfeilschifter, J., L. Bonewald, and G.R. Mundy, *Characterization of the latent transforming growth factor β complex in Bone*. *Journal of Bone and Mineral Research*, 1990. **5**(1): p. 49-58.
106. Kato, M., et al., *Cbfa1-independent decrease in osteoblast proliferation, osteopenia, and persistent embryonic eye vascularization in mice deficient in Lrp5, a Wnt coreceptor*. *The Journal of cell biology*, 2002. **157**(2): p. 303.
107. Yamane, T., et al., *Wnt signaling regulates hemopoiesis through stromal cells*.

The Journal of Immunology, 2001. **167**(2): p. 765.

108. Li, X., et al., *Sclerostin binds to LRP5/6 and antagonizes canonical Wnt signaling*. Journal of Biological Chemistry, 2005. **280**(20): p. 19883.
109. Semënov, M., K. Tamai, and X. He, *SOST is a ligand for LRP5/LRP6 and a Wnt signaling inhibitor*. Journal of Biological Chemistry, 2005. **280**(29): p. 26770.
110. Yoon, B.S. and K.M. Lyons, *Multiple functions of BMPs in chondrogenesis*. Journal of cellular biochemistry, 2004. **93**(1): p. 93-103.
111. Kawabata, M., T. Imamura, and K. Miyazono, *Signal transduction by bone morphogenetic proteins*. Cytokine & growth factor reviews, 1998. **9**(1): p. 49-61.
112. Balint, E., et al., *Phenotype discovery by gene expression profiling: Mapping of biological processes linked to BMP 2 mediated osteoblast differentiation*. Journal of cellular biochemistry, 2003. **89**(2): p. 401-426.
113. Banerjee, C., et al., *Differential regulation of the two principal Runx2/Cbfa1 n-terminal isoforms in response to bone morphogenetic protein-2 during development of the osteoblast phenotype*. Endocrinology, 2001. **142**(9): p. 4026.
114. Lee, K.S., et al., *Runx2 is a common target of transforming growth factor beta 1 and bone morphogenetic protein 2, and cooperation between Runx2 and Smad5 induces osteoblast-specific gene expression in the pluripotent mesenchymal precursor cell line C2C12*. Molecular and cellular biology, 2000. **20**(23): p. 8783.
115. Komori, T., et al., *Targeted disruption of Cbfa1 results in a complete lack of bone formation owing to maturational arrest of osteoblasts*. Cell, 1997. **89**(5): p. 755-764.
116. Phimphilai, M., et al., *BMP Signaling Is Required for RUNX2 Dependent Induction of the Osteoblast Phenotype*. Journal of Bone and Mineral Research, 2006. **21**(4): p. 637-646.
117. Afzal, F., et al., *Smad function and intranuclear targeting share a Runx2 motif*

required for osteogenic lineage induction and BMP2 responsive transcription. Journal of cellular physiology, 2005. **204**(1): p. 63-72.

118. Yang, S., et al., *In Vitro and In Vivo Synergistic Interactions Between the Runx2/Cbfa1 Transcription Factor and Bone Morphogenetic Protein 2 in Stimulating Osteoblast Differentiation.* Journal of Bone and Mineral Research, 2003. **18**(4): p. 705-715.
119. Jilka, R.L., et al., *Increased bone formation by prevention of osteoblast apoptosis with parathyroid hormone.* Journal of Clinical Investigation, 1999. **104**: p. 439-446.
120. Pereira, R.C. and E. Canalis, *Parathyroid hormone increases mac25/insulin-like growth factor-binding protein-related protein-1 expression in cultured osteoblasts.* Endocrinology, 1999. **140**(5): p. 1998.
121. Ambrecht, H., T. Zenser, and B. Davis, *Effect of age on the conversion of 25-hydroxyvitamin D3 to 1, 25-dihydroxyvitamin D3 by kidney of rat.* Journal of Clinical Investigation, 1980. **66**(5): p. 1118.
122. Gallagher, J., et al., *Intestinal calcium absorption and serum vitamin D metabolites in normal subjects and osteoporotic patients: effect of age and dietary calcium.* Journal of Clinical Investigation, 1979. **64**(3): p. 729.
123. ETTINGER, B., H.K. GENANT, and C.E. CANN, *Long-term estrogen replacement therapy prevents bone loss and fractures.* Annals of internal medicine, 1985. **102**(3): p. 319.
124. Ohta, H., et al., *Influence of oophorectomy on serum levels of sex steroids and bone metabolism and assessment of bone mineral density in lumbar trabecular bone by QCT C value.* Journal of Bone and Mineral Research, 1992. **7**(6): p. 659-665.
125. Baron, R., et al., *Processing of undecalcified bone specimens for bone histomorphometry.* Bone Histomorphometry: Techniques and Interpretation. CRC Press, Boca Raton, FL, USA, 1983: p. 13-35.

126. Parfitt, A., *Stereologic basis of bone histomorphometry: Theory of quantitative microscopy and reconstruction of the third dimension*. Bone Histomorphometry: Techniques and Interpretation. CRC Press, Boca Raton, FL, USA, 1983: p. 53–87.
127. Parfitt, A.M., *Stereologic basis of bone histomorphometry: theory of quantitative microscopy and reconstruction of the third dimension*. Bone histomorphometry: techniques and interpretation, 1983: p. 143-223.
128. Howard, C.V. and M.G. Reed, *Unbiased Stereology: Three Dimensional Measurement in Microscopy* 1998, New York, NY: Springer-Verlag.
129. Parfitt, A.M., et al., *Bone histomorphometry: standardization of nomenclature, symbols, and units. Report of the ASBMR Histomorphometry Nomenclature Committee*. J Bone Miner Res, 1987. **2**(6): p. 595-610.
130. Moore, T.L.A. and L.J. Gibson, *Microdamage accumulation in bovine trabecular bone in uniaxial compression*. Journal of biomechanical engineering, 2002. **124**: p. 63.
131. Wang, X. and G.L. Niebur, *Microdamage propagation in trabecular bone due to changes in loading mode*. Journal of biomechanics, 2006. **39**(5): p. 781-790.
132. Dux, S., et al., *Alterations in damage processes in dense cancellous bone following gamma-radiation sterilization*. Journal of biomechanics, 2010. **43**(8): p. 1509-1513.
133. O'Brien, F.J., D. Taylor, and T.C. Lee, *An improved labelling technique for monitoring microcrack growth in compact bone*. J Biomech, 2002. **35**(4): p. 523-6.
134. Burr, D.B. and T. Stafford, *Validity of the bulk-staining technique to separate artifactual from in vivo bone microdamage*. Clin Orthop, 1990. **260**: p. 305-8.
135. Lee, T.C., et al., *Sequential labelling of microdamage in bone using chelating agents*. J Orthop Res, 2000. **18**(2): p. 322-5.

136. Feldkamp, L.A., et al., *The direct examination of three-dimensional bone architecture in vitro by computed tomography*. Journal of Bone and Mineral Research, 1989. **4**(1): p. 3-11.
137. Kuhn, J.L., et al., *Evaluation of a microcomputed tomography system to study trabecular bone structure*. Journal of Orthopaedic Research, 1990. **8**(6): p. 833-842.
138. Hipp, J.A., et al., *Trabecular bone morphology from micro-magnetic resonance imaging*. Journal of Bone and Mineral Research, 1996. **11**(2): p. 286-97.
139. Beck, J.D. and T.M. Keaveny, *A serial-grinding technique for high-resolution imaging of trabecular bone*. Trans. Orthop. Res. Soc., 1996. **21**: p. 709.
140. Salomé, M., et al., *A synchrotron radiation microtomography system for the analysis of trabecular bone samples*. Medical Physics, 1999. **26**: p. 2194.
141. Wang, X., et al., *Detection of trabecular bone microdamage by micro-computed tomography*. Journal of biomechanics, 2007. **40**(15): p. 3397-3403.
142. Tang, S. and D. Vashishth, *A non-invasive in vitro technique for the three-dimensional quantification of microdamage in trabecular bone*. Bone, 2007. **40**(5): p. 1259-1264.
143. Parfitt, A.M., *The physiologic and clinical significance of bone histomorphometric data*, in *Bone histomorphometry: techniques and interpretation*, R.R. Recker, Editor 1983, CRC Press: Boca Raton, FL. p. 143-223.
144. Compston, J., *Bone histomorphometry*, in *Methods in bone biology*, T. Arnett, Editor 1997, Chapman & Hall: London. p. 177-197.
145. Frost, H., *Bone histomorphometry: Choice of marking agent and labeling schedule*. ReckerRR (ed.) Bone Histomorphometry: Techniques and Interpretation. CRC Press, Boca Raton, FL, 1983: p. 37-52.

146. Ott, S.M., *Histomorphometric analysis of bone remodeling*. Principles of bone biology, 2002: p. 303.
147. Frost, H.M., *Tetracycline-based histological analysis of bone remodeling*. Calcif Tissue Res, 1969. **3**(3): p. 211-37.
148. Kimmel, D.B. and W.S.S. Jee, *Measurements of Area, Perimeter, and Distance: Details of Data Collection in Bone Histomorphometry*, in *Bone Histomorphometry: Techniques and Interpretation* 1983. p. 90-107.
149. Rao, D.S., *Practical Approach to Bone Biopsy*, in *Bone Histomorphometry: Techniques and Interpretation* 1983. p. 3-11.
150. An, Y.H.M., K.L., *Handbook of Histology Methods for Bone and Cartilage* 2003, Totowa, New Jersey: Humana Press.
151. Recker, R., *Introduction to bone histomorphometry*, in *Bone histomorphometry : techniques and interpretation*, R. Recker, Editor 1983, CRC Press: Boca Raton, Fla.
152. Hauge, E.M., et al., *How many patients are needed? Variation and design considerations in bone histomorphometry*. Bone, 2001. **28**(5): p. 556-62.
153. Ehlert, K., R. O'Brien, and C. Hernandez. *Statistical Power in Measures of Microscopic Tissue Damage in Cancellous Bone*. in *Am Soc Bone Miner Res*. 2011.
154. Mouton, P.R., *Principles and practices of unbiased stereology* 2002, Baltimore, MD, USA: Johns Hopkins University Press.
155. Hauge, E.M., L. Mosekilde, and F. Melsen, *Stereological considerations concerning the measurement of individual osteoid seams and resorption cavities*. J Bone Miner Res, 1994. **26**(1): p. 89-90.
156. Nagaraja, S., T.L. Couse, and R.E. Guldberg, *Trabecular bone microdamage and microstructural stresses under uniaxial compression*. Journal of biomechanics,

2005. **38**(4): p. 707-716.

- 157. O'Neal, J.M., et al., *One year of alendronate treatment lowers microstructural stresses associated with trabecular microdamage initiation*. Bone, 2010. **47**(2): p. 241-247.
- 158. Qiu, S., et al., *The morphological association between microcracks and osteocyte lacunae in human cortical bone*. Bone, 2005. **37**(1): p. 10-15.
- 159. Compston, J.E., *Bone histomorphometry -- The Renaissance?* BoneKey-Osteovision, 2004. **1**(5): p. 9-12.
- 160. Weibel, E.R., *Stereological Methods*. 1979.
- 161. Compston, J.E.V., S.; Stellon, A.J., *Inter-observer and Intra-observer Variation in Bone Histomorphometry*. Calcified Tissue International, 1986. **38**(2): p. 67-70.
- 162. Wright, C.D.P., et al., *Combined inter-observer and inter-method variation in bone histomorphometry*. Bone, 1992. **13**(3): p. 205-208.
- 163. Vesterby, A., et al., *Unbiased stereologic estimation of surface density in bone using vertical sections*. Bone, 1987. **8**(1): p. 13-7.
- 164. Odgaard, A., et al., *A direct method for fast three-dimensional serial reconstruction*. J Microsc, 1990. **159**: p. 335-342.
- 165. Beck, J.D., et al., *Three-dimensional imaging of trabecular bone using the computer numerically controlled milling technique*. Bone, 1997. **21**(3): p. 281-7.
- 166. Kazakia, G.J., et al., *Automated high-resolution three-dimensional fluorescence imaging of large biological specimens*. J Microsc, 2007. **225**(Pt 2): p. 109-17.
- 167. Wu, H.S.B., J.; Gil, J. , *Iterative Thresholding for Segmentation of Cells from Noisy Images*. Journal of Microscopy, 2001. **197**(3): p. 296-304.

168. Tomazevic, D.L., B.; Pernus, F., *Comparative Evaluation of Retrospective Shading Correction Methods*. Journal of Microscopy, 2002. **208**(3): p. 212-223.
169. Chow, S.K., et al., *Automated microscopy system for mosaic acquisition and processing*. J Microsc, 2006. **222**(Pt 2): p. 76-84.
170. Russ, J.C., *The Image Processing Handbook*. fifth eddition ed2006: CRC Press. 817.
171. Waarsing, J., J. Day, and H. Weinans, *An Improved Segmentation Method for In Vivo uCT Imaging*. Journal of Bone and Mineral Research, 2004. **19**(10): p. 1640-50.
172. Burghardt, A.J.K., G.J.; Majumdar, S., *A Local Adaptive Threshold Strategy for High Resolution Peripheral Quantitative Computed Tomography of Trabecular Bone*. Annals of Biomedical Engineering, 2007. **35**(10): p. 1678-86.
173. Zack, G.W., W.E. Rogers, and S.A. Latt, *Automatic measurement of sister chromatid exchange frequency*. J. Histochem. Cytochem., 1977. **25**(7): p. 741-753.
174. Kazakia, G.J., et al., *Assessment of bone tissue mineralization by conventional x-ray microcomputed tomography: Comparison with synchrotron radiation microcomputed tomography and ash measurements*. Medical Physics, 2008. **35**(7): p. 3170-3179.
175. Mosekilde, L., *Consequences of the remodelling process for vertebral trabecular bone structure: a scanning electron microscopy study (uncoupling of unloaded structures)*. Bone Miner, 1990. **10**(1): p. 13-35.
176. Li, C.Y., et al., *Estrogen and "Exercise" Have a Synergistic Effect in Preventing Bone Loss in the Lumbar Vertebra and Femoral Neck of the Ovariectomized Rat*. Calcified Tissue International, 2003. **72**: p. 42-49.
177. Parfitt, A.M., et al., *A new model for the regulation of bone resorption, with particular reference to the effects of bisphosphonates*. J Bone Miner Res, 1996.

11(2): p. 150-9.

178. Seeman, E. and P.D. Delmas, *Bone Quality — The Material and Structural Basis of Bone Strength and Fragility*. New Eng J Med, 2006. **354**(21): p. 2250-2261.
179. Hernandez, C.J., *How can bone turnover modify bone strength independent of bone mass?* Bone, 2008. **42**(6): p. 1014-1020.
180. Kummari, S., T. Coan, and C. Hernandez. *Stress Concentration Factors of Remodeling Cavities on Rod-like and Plate-like Trabeculae*. in Orth Res Soc. 2011.
181. Slyfield, C.R., Jr., et al., *Three-dimensional surface texture visualization of bone tissue through epifluorescence-based serial block face imaging*. J Microsc, 2009. **236**(1): p. 52-9.
182. Tkachenko, E.V., et al., *Voxel size and measures of individual resorption cavities in three-dimensional images of cancellous bone*. Bone, 2009. **45**(3): p. 487-92.
183. Li, C.Y., et al., *Estrogen and "exercise" have a synergistic effect in preventing bone loss in the lumbar vertebra and femoral neck of the ovariectomized rat*. Calcif Tissue Int, 2003. **72**(1): p. 42-9.
184. Jee, W.S.S., et al., *Maintaining restored bone with bisphosphonate in the ovariectomized rat skeleton: Dynamic histomorphometry of changes in bone mass*. Bone, 1993. **14**(3): p. 493-498.
185. Riggs, B.L. and A.M. Parfitt, *Drugs Used to Treat Osteoporosis: The Critical Need for a Uniform Nomenclature Based on Their Action on Bone Remodeling*. J Bone Miner Res, 2005. **20**(2): p. 177-184.
186. Parfitt, A.M., *Skeletal Heterogeneity and the Purposes of Bone Remodeling: Implications for the Understanding of Osteoporosis*, in Osteop, R. Marcus, et al., Editors. 2008, Academic Press.
187. Wronski, T.J., et al., *Long-term effects of ovariectomy and aging on the rat*

- skeleton*. Calcif Tiss Int, 1989. **45**(6): p. 360-366.
188. Parfitt, A.M., *Trabecular bone architecture in the pathogenesis and prevention of fracture*. Am J Med, 1987. **82**(1B): p. 68-72.
 189. van der Linden, J.C., J.A. Verhaar, and H. Weinans, *A three-dimensional simulation of age-related remodeling in trabecular bone*. J Bone Miner Res, 2001. **16**(4): p. 688-96.
 190. Steyer, G.J., et al., *Removal of out-of-plane fluorescence for single cell visualization and quantification in cryo-imaging*. Ann Biomed Eng, 2009. **37**(8): p. 1613-1628.
 191. Evans, G., et al., *The effects of raloxifene on tibia histomorphometry in ovariectomized rats*. Endoc, 1994. **134**(5): p. 2283-2288.
 192. Recker, R.R., *Bone histomorphometry: techniques and interpretation* 1983, Boca Raton, Fla.: CRC Press.
 193. Bouxsein, M.L., et al., *Guidelines for assessment of bone microstructure in rodents using micro-computed tomography*. J Bone Miner Res, 2010. **25**(7): p. 1468-1486.
 194. Eriksen, E.F., F. Melsen, and L. Mosekilde, *Reconstruction of the resorptive site in iliac trabecular bone: a kinetic model for bone resorption in 20 normal individuals*. Metab Bone Dis Relat Res, 1984. **5**(5): p. 235-42.
 195. Kopperdahl, D.L., E.F. Morgan, and T.M. Keaveny, *Quantitative computed tomography estimates of the mechanical properties of human vertebral trabecular bone*. J Orth Res, 2002. **20**(4): p. 801-805.
 196. Bagi, C.M., et al., *Benefit of systemically administered rhIGF I and rhIGF I/IGFBP 3 on cancellous bone in ovariectomized rats*. J Bone Miner Res, 1994. **9**(8): p. 1301-1312.
 197. Dobnig, H. and R.T. Turner, *Evidence that intermittent treatment with*

parathyroid hormone increases bone formation in adult rats by activation of bone lining cells. Endoc, 1995. **136**(8): p. 3632-8.

198. Schmidt, I., H. Dobnig, and R. Turner, *Intermittent parathyroid hormone treatment increases osteoblast number, steady state messenger ribonucleic acid levels for osteocalcin, and bone formation in tibial metaphysis of hypophysectomized female rats.* Endoc, 1995. **136**(11): p. 5127.
199. Meng, X.W., et al., *Temporal expression of the anabolic action of PTH in cancellous bone of ovariectomized rats.* J Bone Miner Res, 1996. **11**(4): p. 421-429.
200. Frost, H.M., *Bone histomorphometry: Correction of the labelling "escape error."*, in ReckerRR (ed.) *Bone Histomorphometry: Techniques and Interpretation.* CRC Press, Boca Raton, FL1983. p. 133–142.
201. Bell, K.L., et al., *Super-osteons (remodeling clusters) in the cortex of the femoral shaft: Influence of age and gender.* Anat Rec, 2001. **264**(4): p. 378-386.
202. Jilka, R., et al., *Increased osteoclast development after estrogen loss: mediation by interleukin-6.* Science, 1992. **257**(5066): p. 88.
203. Jilka, R.L., et al., *Estrogen loss upregulates hematopoiesis in the mouse: a mediating role of IL-6.* Exp Hemat, 1995. **23**(6): p. 500.
204. Kimble, R.B., et al., *Estrogen deficiency increases the ability of stromal cells to support murine osteoclastogenesis via an interleukin-1 and tumor necrosis factor-mediated stimulation of macrophage colony-stimulating factor production.* J Bio Chem, 1996. **271**(46): p. 28890.
205. Romas, E., et al., *The role of gp130-mediated signals in osteoclast development: Regulation of interleukin 11 production by osteoblasts and distribution of its receptor in bone marrow cultures.* J Exp Med, 1996. **183**(6): p. 2581-2591.
206. Nakamura, T., et al., *Estrogen prevents bone loss via estrogen receptor alpha and induction of Fas ligand in osteoclasts.* Cell, 2007. **130**(5): p. 811-823.

207. Parfitt, A.M., et al., *Theoretical perspective: A new model for the regulation of bone resorption, with particular reference to the effects of bisphosphonates*. J Bone Miner Res, 1996. **11**(2): p. 150-159.
208. Garnero, P., *Markers of bone turnover for the prediction of fracture risk*. Osteoporos Int, 2000. **11 Suppl 6**: p. S55-65.
209. Heaney, R.P., *Is the paradigm shifting?* Bone, 2003. **33**(4): p. 457-65.
210. Dempster, D.W., *The contribution of trabecular architecture to cancellous bone quality*. Journal of Bone and Mineral Research, 2000. **15**(1): p. 20-23.
211. Parfitt, A.M., *High bone turnover is intrinsically harmful: two paths to a similar conclusion. The Parfitt view*. J Bone Miner Res, 2002. **17**(8): p. 1558-9; author reply 1560.
212. Smit, T.H. and E.H. Burger, *Is BMU-coupling a strain-regulated phenomenon? A finite element analysis*. J Bone Miner Res, 2000. **15**(2): p. 301-7.
213. Eswaran, S.K., et al., *A computational assessment of the independent contribution of changes in canine trabecular bone volume fraction and microarchitecture to increased bone strength with suppression of bone turnover*. Journal of biomechanics, 2007. **40**(15): p. 3424-3431.
214. Morgan, E.F., et al., *Contribution of inter-site variations in architecture to trabecular bone apparent yield strains*. J Biomech, 2004. **37**(9): p. 1413-20.
215. Bigley, R., et al., *Validity of serial milling-based imaging system for microdamage quantification*. Bone, 2008. **42**(1): p. 212-215.
216. Bevil, G., et al., *The influence of boundary conditions and loading mode on high-resolution finite element-computed trabecular tissue properties*. Bone, 2009. **44**(4): p. 573-578.
217. Rasband, W., *ImageJ*, 1997-2011, National Institutes of Health: Bethesda, MD,

USA.

- 218. Doube, M., et al., *BoneJ: Free and Extensible Bone Image Analysis in ImageJ*. Bone, 2010. **47**: p. 1076-1079.
- 219. Nagaraja, S., A.S.P. Lin, and R.E. Guldberg, *Age-related changes in trabecular bone microdamage initiation*. Bone, 2007. **40**(4): p. 973-980.
- 220. Tang, S.Y. and D. Vashishth, *Non-enzymatic glycation alters microdamage formation in human cancellous bone*. Bone, 2010. **46**(1): p. 148-154.
- 221. Karim, L. and D. Vashishth, *Role of trabecular microarchitecture in the formation, accumulation, and morphology of microdamage in human cancellous bone*. J Ortho Res, 2011.
- 222. Wang, X., et al., *Axial-shear interaction effects on microdamage in bovine tibial trabecular bone*. European journal of morphology, 2005. **42**(1/2): p. 61-70.
- 223. Staa, T.P., H. Leufkens, and C. Cooper, *The epidemiology of corticosteroid-induced osteoporosis: a meta-analysis*. Osteoporosis International, 2002. **13**(10): p. 777-787.
- 224. Pitsillides, A.A., et al., *Mechanical strain-induced NO production by bone cells: a possible role in adaptive bone (re)modeling?* Faseb Journal, 1995. **9**(15): p. 1614-22.
- 225. Lanyon, L.E., *Control of bone architecture by functional load bearing*. Journal of Bone and Mineral Research, 1992: p. S369-75.
- 226. Manolagas, S., *Choreography from the tomb: an emerging role of dying osteocytes in the purposeful, and perhaps not so purposeful, targeting of bone remodeling*. BoneKEy, 2006. **3**: p. 5-14.
- 227. el Haj, A.J., et al., *Cellular responses to mechanical loading in vitro*. Journal of Bone and Mineral Research, 1990. **5**(9): p. 923-32.

228. Lean, J.M., et al., *Increased insulin-like growth factor I mRNA expression in response to mechanical stimulation*. Am J Physiol, 1995. **268**(2 Pt 1): p. E318-27.
229. Kamioka, H., et al., *Terminal Differentiation of Osteoblasts to Osteocytes Is Accompanied by Dramatic Changes in the Distribution of Actin Binding Proteins*. Journal of Bone and Mineral Research, 2004. **19**(3): p. 471-478.
230. Doty, S.B., *Morphological evidence of gap junctions between bone cells*. Calcified Tissue International, 1981. **33**: p. 509-512.
231. Donahue, H.J., et al., *Cell-to-Cell Communication In Osteoblastic Networks - Cell Line-Dependent Hormonal Regulation Of Gap Junction Function*. Journal of Bone and Mineral Research, 1995. **10**(6): p. 881-889.
232. Tomkinson, A., et al., *The death of osteocytes via apoptosis accompanies estrogen withdrawal in human bone*. Journal of Clinical Endocrinology & Metabolism, 1997. **82**(9): p. 3128.
233. Rodan, G.A., *Perspectives: mechanical loading, estrogen deficiency, and the coupling of bone formation to bone resorption*. Journal of Bone and Mineral Research, 1991. **6**(6): p. 527-530.
234. Busse, B., et al., *Decrease in the osteocyte lacunar density accompanied by hypermineralized lacunar occlusion reveals failure and delay of remodeling in aged human bone*. Aging Cell, 2010. **9**(6): p. 1065-1075.
235. Weinstein, R.S., R.W. Nicholas, and S.C. Manolagas, *Apoptosis of osteocytes in glucocorticoid-induced osteonecrosis of the hip*. Journal of Clinical Endocrinology & Metabolism, 2000. **85**(8): p. 2907.
236. van der Meulen, M.C.H., et al., *Cancellous bone adaptation to in vivo loading in a rabbit model*. Bone, 2006. **38**(6): p. 871-877.
237. McNamara, L.M., et al., *Stress-concentrating effect of resorption lacunae in trabecular bone*. J Biomech, 2006. **39**(4): p. 734-41.

238. Boyde, A. and M. Hobdell, *Scanning electron microscopy of bone*. Calcified Tissue Research, 1968: p. Suppl:4-4B.
239. Jones, S.J. and A. Boyde, *Histomorphometry of Howship's lacunae formed in vivo and in vitro: depths and volumes measured by scanning electron and confocal microscopy*. Bone, 1993. **14**(3): p. 455-60.
240. Ito, M., et al., *Bone structure and mineralization demonstrated using synchrotron radiation computed tomography (SR-CT) in animal models: preliminary findings*. J Bone Miner Metab, 2003. **21**(5): p. 287-93.
241. Eswaran, S.K., et al., *A computational assessment of the independent contribution of changes in canine trabecular bone volume fraction and microarchitecture to increased bone strength with suppression of bone turnover*. J Biomech, 2007. **40**(15): p. 3424-31.
242. Slyfield, C.R., et al., *3D surface texture visualization of bone tissue through epi-fluorescence-based serial block face imaging*. Submitted, 2008.
243. Slyfield, C.R., et al. *3D visualization and measurement of resorption cavities in cancellous bone*. in *Transactions of the Orthopaedic Research Society*. 2008. San Francisco, CA, USA: Orthopaedic Research Society.
244. Russ, J.C., *The Image Processing Handbook*. 5th ed2007, Boca Raton, FL, USA: CRC Press.
245. Garrahan, N.J., P.I. Croucher, and J.E. Compston, *A computerised technique for the quantitative assessment of resorption cavities in trabecular bone*. Bone, 1990. **11**(4): p. 241-5.
246. Peyrin, F., et al., *Micro-CT examinations of trabecular bone samples at different resolutions: 14, 7 and 2 micron level*. Technol Health Care, 1998. **6**(5-6): p. 391-401.
247. Hernandez, C.J. and T.M. Keaveny, *A biomechanical perspective on bone quality*. Bone, 2006. **39**(6): p. 1173-81.